NEW RASPBERRY PI ZERO W OUT NOW!

The MagPi

The official Raspberry Pi magazine

Issue 55 March 2017

raspberrypi.org/magpi

BUILD A DIGITAL WHOOPEE CUSHION

Have a laugh with physical computing

MAKE AN ALEXA ROBOT

Take voice control with Amazon's Alexa

RETRO GAMING BLISS

Build this handheld gaming masterpiece with the brand new **Raspberry Pi Zero W**

BACK UP YOUR RASPBERRY PI

Keep your files safe

ADVANCED GPIO CONTROL

Improve your projects

PI ZERO NOW HAS WIRELESS!

And it's hidden inside this month's big-build project

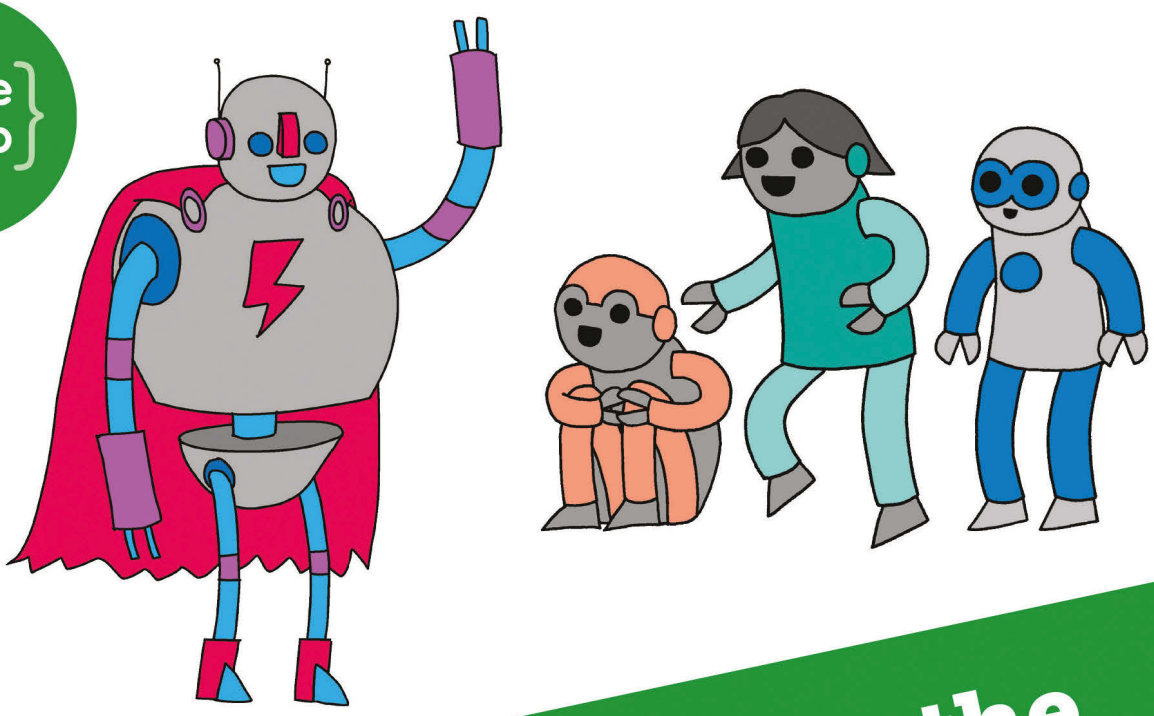
Also inside:

- THE TINY PI-POWERED SMARTPHONE
- UPGRADE YOUR TV WITH A PI HTPC
- 3D MODELLING ON RASPBERRY PI
- KODI 17 REVIEWED AND RATED

Issue 55 • Mar 2017 • £5.99

9 772051 998001 03

THE ONLY MONTHLY MAGAZINE WRITTEN BY AND FOR THE PI COMMUNITY



Can you help inspire the next generation of coders?

Code Club is a nationwide network of volunteer-led after school clubs for children aged 9-11.

We're always looking for people with coding skills to volunteer to run a club at their local primary school, library or community centre for an hour a week.

You can team up with colleagues, a teacher will be there to support you and we provide all the materials you'll need to help get children excited about digital making.

There are loads of ways to get involved!

So to find out more, join us at **www.codeclub.org.uk**

This advertising space was kindly donated to Code Club by MagPi Magazine

WELCOME TO THE OFFICIAL MAGAZINE

It's been extremely hard to keep the existence of the Pi Zero W secret over the last few months. When Eben Upton first revealed that he had something new in the pipeline, we spent the rest of the day discussing how much of a game-changer it could be for hackers and makers around the world.

Now that the wireless connectivity of the Pi Zero W has been revealed, you can find the definitive breakdown – including some background into its making – starting on page 64. And beginning on page 14, this month's big build integrates the new Pi model with Adafruit's amazing PiGRRL handheld console, making it even better than before.

To celebrate even further, we've got a new subscription offer that lets you walk away with your own Pi Zero W and brand-new official Pi Zero case that comes with three interchangeable covers. You'll also get cables for the Camera Module, HDMI, and USB ports. Learn more on page 28. Enjoy the issue!

Rob Zwetsloot
Features Editor

THIS MONTH:

14 BUILD THE ULTIMATE HANDHELD
Upgrade the famous PiGRRL console in this month's big build

42 MAKE AN ALEXA-POWERED ROBOT
This Alexabot responds to your emotions

64 THE PI ZERO W IS HERE!
The latest version of the Pi Zero is amazing. Here's why

74 KODI 17 REVIEWED
The most popular HTPC software in the world gets better

FIND US ONLINE raspberrypi.org/magpi **GET IN TOUCH** magpi@raspberrypi.org

The MagPi

Available on the App Store

GET IT ON Google Play

EDITORIAL
Managing Editor: **Russell Barnes** russell@raspberrypi.org
Features Editor: **Rob Zwetsloot**
News Editor: **Lucy Hattersley**
Sub Editors: **Laura Clay, Phil King, Lorna Lynch**

DESIGN
Critical Media: **criticalmedia.co.uk**
Head of Design: **Dougal Matthews**
Designers: **Lee Allen, Daiva Bumelyte, Mike Kay**
Illustrator: **Sam Alder**

PUBLISHING
For advertising & licensing:
russell@raspberrypi.org +44 (0)7904 766523
Publisher: **Liz Upton**
CEO: **Eben Upton**

DISTRIBUTION
Seymour Distribution Ltd
2 East Poultry Ave
London
EC1A 9PT | **+44 (0)207 429 4000**

SUBSCRIPTIONS
Select Publisher Services Ltd
PO Box 6337
Bournemouth
BH1 9EH | **+44 (0)1202 586 848**

CONTRIBUTORS
Sam Aaron, Alex Bate, Jamie Bailey, Germano Cesari, Rachel Churcher, John Cole, Phil King, Simon Long, Carrie Anne Philbin, Ben Nuttall, Matt Richardson, and Richard Smedley

recycle
When you have finished with this magazine, please recycle it.

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

The MagPi magazine is published by Raspberry Pi (Trading) Ltd., 30 Station Road, Cambridge, CB1 2JH. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2051-9982.

Contents

Issue 55 March 2017 raspberrypi.org/magpi

TUTORIALS

> PI 101 – BUILD A HTPC **38**
Make the ultimate Kodi media centre

> PI 101 – BACK UP YOUR PI **40**
Keep your important files safe

> VOICE-CONTROLLED ROBOT **42**
Build an Alexa-powered Pi robot

> INTRO TO C 9 – USER INPUTS **46**
Get user input and implement it in your C code

> MAKE 3D MODELS ON YOUR PI **48**
Create a squid in Glise on Raspberry Pi

> SONIC PI SOUND DESIGN **52**
Make new sounds by removing synth elements

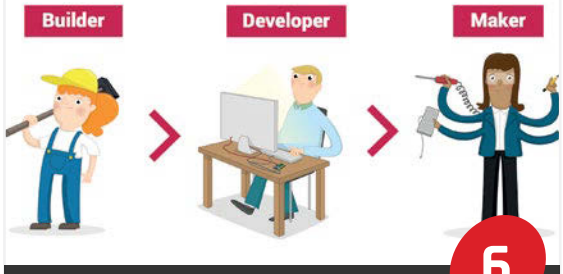
> ADVANCED GPIO ZERO **54**
Use Source/Values to make your code more efficient

> MAKE A WHOOPI CUSHION **56**
Create an electronic farting machine

> REMOTELY MONITOR PROCESSES **60**
Create a process dashboard with email alerts

IN THE NEWS

PI CURRICULUM



New school curriculum uses digital making 6

COVER FEATURE

14

BUILD A RETRO HANDHELD

COUNTERFEIT CASES

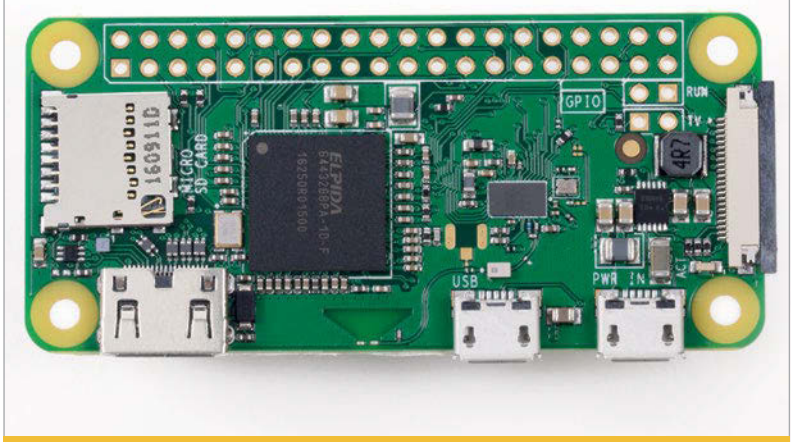
Don't get caught out by these fake Pi cases 10

GOOGLE SURVEY

Google wants to know if Pi users want AI tools 13

Contents

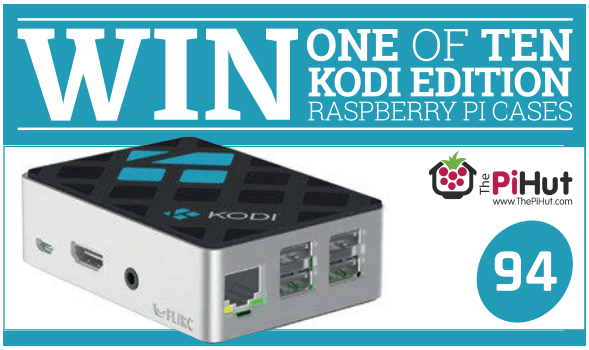
THE BIG FEATURE



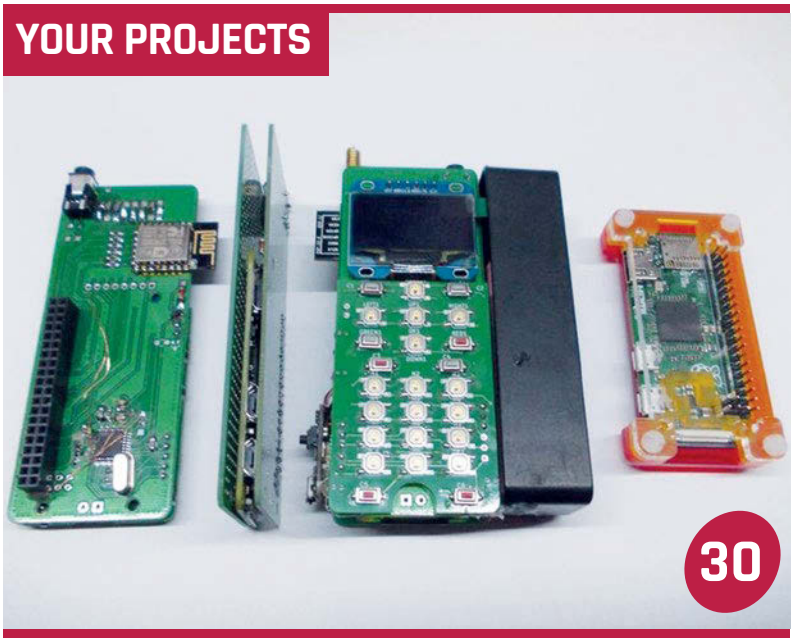
PI ZERO W

You asked for wireless LAN. You got wireless LAN. Here's everything you need to know about the improved Pi Zero

64



YOUR PROJECTS



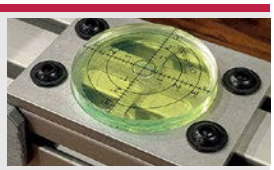
ZEROPHONE

A Raspberry Pi Zero that's also a phone

PI TELESCOPE

32

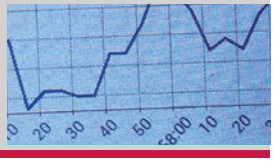
View the stars with Pi-aided accuracy



HEARTBEAT MONITOR

34

A fitness project that monitors your heart rate



REGULARS

> NEWS 06
> TECHNICAL FAQ 62
> BOOK REVIEWS 80
> FINAL WORD 96

COMMUNITY

> THIS MONTH IN PI 84
What else happened in the world of Pi?
> COMMUNITY SPOTLIGHT 86
This month it's time for some Pi and Chips
> CODE CLUB INTERNATIONAL 88
How Code Club is helping kids globally
> EVENTS 90
Find a community event near you
> LETTERS 92
We answer your pressing questions about the mag

REVIEWS

> KODI 17 74
> PI METRE 76
> SPEAKER PHAT 77
> CLUSTER HAT 78

RASPBERRY PI CREATES CURRICULUM

New curriculum aims to teach the power of digital making

The Raspberry Pi Foundation is determined to put the power of digital making into the hands of people all over the world.

It has taken an exciting step in this journey, by defining its digital making curriculum that will help people everywhere learn new skills.

You can view the new curriculum online (**magpi.cc/2kZD18j**).

For the community

"We have a large and diverse community of people who are interested in digital making," explains Carrie Anne Philbin, director of education at the Raspberry Pi Foundation.

Some of those people might use the curriculum to help guide and inform their learning, or perhaps their children's education.

Individuals who run digital making clubs at schools, community centres, and Raspberry Jams may draw on it for extra guidance.

Some teachers may wish to use the curriculum as inspiration for what to teach their students.

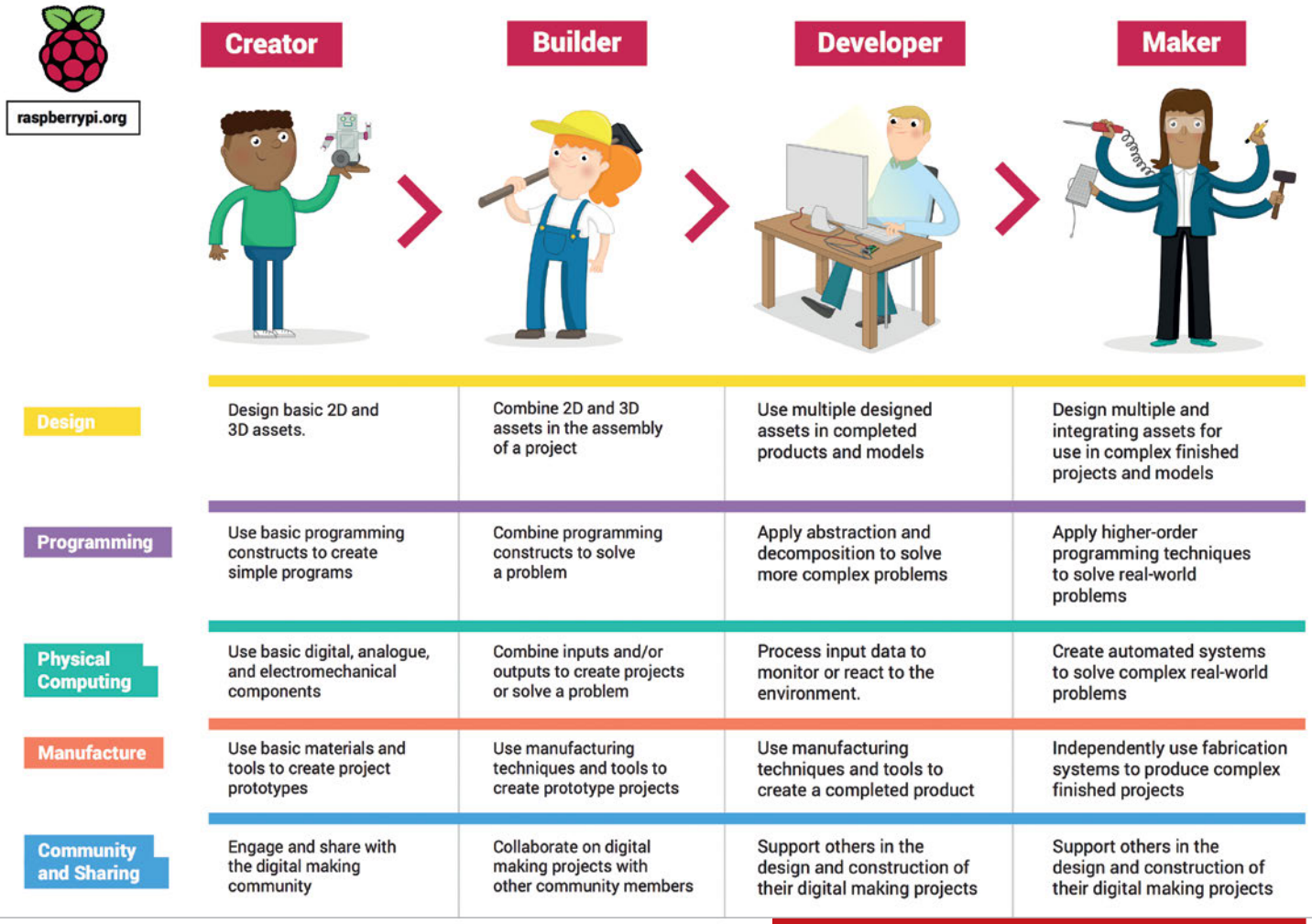
"We want to help people to make things so that they can become the inventors, creators, and makers of tomorrow," says

Below Teachers can use the curriculum to develop their skills and as inspiration for school lessons

HOW TO GET INVOLVED

The Raspberry Pi Foundation is proud to be part of the maker movement. It is looking for feedback from the community on its new curriculum.

Please add your feedback to this form: **magpi.cc/2kKfQ3e**.


raspberrypi.org
Creator
Builder
Developer
Maker
Design
Design basic 2D and 3D assets.
Combine 2D and 3D assets in the assembly of a project
Use multiple designed assets in completed products and models
Design multiple and integrating assets for use in complex finished projects and models
Programming
Use basic programming constructs to create simple programs
Combine programming constructs to solve a problem
Apply abstraction and decomposition to solve more complex problems
Apply higher-order programming techniques to solve real-world problems
Physical Computing
Use basic digital, analogue, and electromechanical components
Combine inputs and/or outputs to create projects or solve a problem
Process input data to monitor or react to the environment.
Create automated systems to solve complex real-world problems
Manufacture
Use basic materials and tools to create project prototypes
Use manufacturing techniques and tools to create prototype projects
Use manufacturing techniques and tools to create a completed product
Independently use fabrication systems to produce complex finished projects
Community and Sharing
Engage and share with the digital making community
Collaborate on digital making projects with other community members
Support others in the design and construction of their digital making projects
Support others in the design and construction of their digital making projects

Carrie Anne. Digital making, STEAM, project-based learning and tinkering are at the core of the Foundation's teaching philosophy. This principle can be summed up simply as "we learn best by doing," Carrie Anne tells us.

The curriculum

The curriculum has five strands: Design, Programming, Physical Computing, Manufacture, and Community and Sharing. Each of these encapsulates key concepts and skills in digital making. Users will progress in each strand. "You start with certain skills and knowledge," says Carrie Anne. Then, with guidance, practice, and understanding, you progress towards broader and deeper knowledge and competence.

Characters are used to represent different levels of ability. Progression starts with Creator, and moves onto Builder and Developer before becoming Maker.

Learning for all

The Raspberry Pi Curriculum is open to everybody, regardless of age or experience. "We wanted to avoid the age-related and stage-related labels that are often associated with a learner's progress," reveals Carrie Anne, "[along with] the preconceptions, these labels bring."

The Raspberry Pi Foundation's digital making curriculum is not intended to be a replacement for computer science-related curricula around the world.

It hopes that following the learning pathways will support the study of formal curricular and exam specifications in a fun and tangible way.

The Foundation is continuing to expand its catalogue of free educational resources. It expects the curriculum to grow and improve, but is requesting input into that process. This is "vital" says Carrie Anne. So please do take a closer look at the program (see 'Access the curriculum') and send your feedback (see 'How to get involved').

> "We want to help people to make things so that they can become the inventors, creators, and makers of tomorrow"

ACCESS THE CURRICULUM

A PDF version of the curriculum is available to download direct from the Raspberry Pi Foundation: **magpi.cc/2kZzvu0**.

HIGH SCHOOL REPLACES COMPUTERS WITH RASPBERRY PI

MC2 STEM High School using Raspberry Pi to boost computer science skills

One school in Cleveland, Ohio has chosen to use Raspberry Pi rather than buy expensive new computers.

Instead of the Raspberry Pi being a limitation, the smaller boards have enabled MC2 STEM to implement a full computer science curriculum.

"MC2 STEM High School is a unique school," says Feowyn A MacKinnon, head of school.

There are 50,000 students in the district, and 100% of them qualify for free and reduced meals. As a result, the technology situation can be quite bleak.

The computers at MC2 were ten years old and hopelessly out of date. "More useful as paperweights than technology," Feowyn tells us.

A mobile Fab Lab enables MC2 STEM students to build projects and demonstrate them to nearby schools

> "The Raspberry Pi is better for teaching STEM than traditional equipment"

Traditionally, schools in America would have a fund-raising drive to try to secure the money required for new technology. However, MC2 had a better idea.

"Rather than try to fund-raise for new technology," explains Feowyn, "we have come to depend completely on Raspberry Pi."

It turns out the Raspberry Pi is better for teaching STEM than traditional equipment. "Our students experience an innovative and fun project while building their Raspberry Pi computers," reveals Feowyn.

MC2 is now teaching a full year-long computer science curriculum using only Raspberry Pi computers. "I imagine we may be one of the only high schools doing this," says Feowyn.

"The Raspberry Pi then evolved from just a cool, fun, learning experience to an all-encompassing educational tool that our school and curriculum truly depend on," he continues.

MC2 is also the first high school in the US to own a mobile Fab Lab. "The mobile Fab Lab comes in handy when we want to express our ideas to other schools," says MC2 STEM student Brandon (**magpi.cc/2kLla6s**). "I like the idea of project-based learning and being able to build stuff with your hands, instead of always doing writing work."

ONLY
$109.99

CREATE SOMETHING NEW

WITH WD PIDRIVE COMPUTE CENTRE

Includes custom NOOBS, Raspbian PIXEL, and Raspbian Lite on a bundled SD card. Organize multiple projects in one place with Project Spaces™. Easily assemble components in a custom enclosure.

Kit Includes:
WD PiDrive Foundation Edition 375GB
Raspberry Pi 3
Wireless Keyboard & Mouse
Black Square 6"x6" Enclosure
WD PiDrive Cable
5V 3A power supply and USB cable
microSD card (with preloaded software)

WDLABS™

Learn more at **wdlabs.wd.com/magpi55a**

Western Digital, WD, the WD logo, WDLabs, Project Spaces, and the WDLabs logo are registered trademarks or trademarks of Western Digital Corporation or its affiliates in the U.S. and/or other countries. Raspberry Pi is a trademark of the Raspberry Pi Foundation. microSD is a trademark of SD-3C, LLC. All other marks are the property of their respective owners. As used for storage capacity, one gigabyte = one billion bytes. Total accessible capacity varies depending on operating environment. Product specifications subject to change without notice. Pictures shown may vary from actual product. © 2017 Western Digital Corporation or its affiliates.

WATCH OUT FAKE CASES

Above This fake case has two sets of light pipes
Below The shoddy feet on this case are a clear sign of a fake

Fake Raspberry Pi cases doing the rounds

Unscrupulous dealers are selling fake Raspberry Pi cases on third-party websites and eBay.

Kinneir Dufort designed the official Raspberry Pi case. While there are many third-party cases available, the fake cases are intended to mimic the official case.

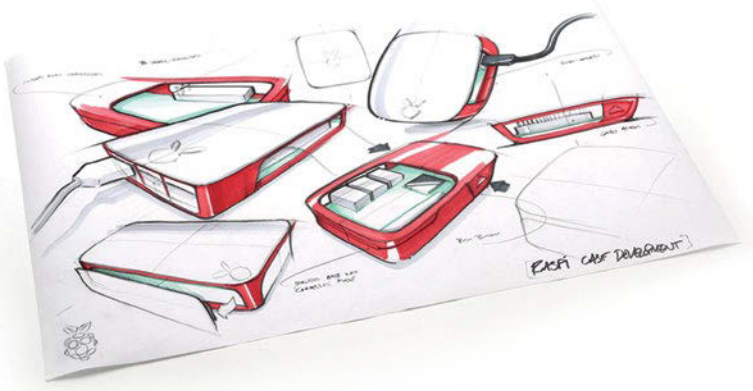
The Raspberry Pi Foundation managed to get its hands on a sample through a proxy pretending to be a Pi shop. It supplied some pictures so the community can see the differences.

"The fake cases are not as well made as the real thing," says Liz Upton, director of communications. "They also deprive us of some much-needed charitable income. As you probably know, the Raspberry Pi Foundation is a charity. All the money we make from selling computers, cases, cameras, and other products goes straight into our charitable fund to train teachers, provide free learning resources, teach kids, help build the foundations of digital making in schools, and much more."

Let The Raspberry Pi Foundation know if you find fake cases in the wild. "It damages our reputation when something shonky comes on the market and it looks like we've made it," explains Liz. "It damages the Raspberry Pi Foundation's pockets too, which means we can't do the important work in education we were set up to do."

> "It damages our reputation when something shonky comes on the market"

You can see examples of fake versus real cases on the Raspberry Pi Foundation website (**magpi.cc/2kLw3pf**). Please let them know in the comments if you come across any fakes.



HELLO WORLD

New free magazine for educators

The Raspberry Pi Foundation has launched a new magazine called *Hello World* (**magpi.cc/2j4HuJ9**).

Hello World is a magazine about computing and digital making. It's "written by educators, for educators," says the Foundation.

Hello World is designed to be cross-curricular and useful to all kinds of educators, from classroom teachers to librarians. While it includes lots of great examples of how teachers are using Raspberry Pi computers in education, it is device- and platform-neutral.

"As with everything we do at the Raspberry Pi Foundation, *Hello World* is about community building," says Raspberry Pi Foundation CEO, Philip Colligan. "Our goal is to provide a resource that will help educators connect, share great practice, and learn from each other."

Hello World is available free, forever, for everyone online as a downloadable PDF. The content is written to be internationally relevant, and includes features on the most exciting developments and best practices from around the world.

A print version of the magazine is free to the homes of serving educators in the UK. The list of eligible subscribers is wider than you think and includes teachers, teaching assistants, librarians, and youth workers. Alternatively, you can subscribe for £15 per year.

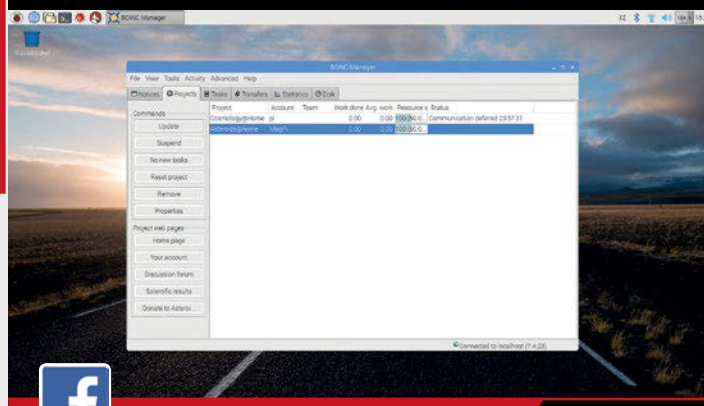
Visit the *Hello World* website to sign up for your free subscription (**magpi.cc/2loKN1R**).

(Hello World)
WELCOME TO THE MAGAZINE
NEW!
JOIN THE CLUB
CODE CLUB CO-FOUNDER CLARE SUTCLIFFE SHARES HER THOUGHTS ON BREAKING THE 5,000 UK CLUBS MILESTONE, WORLDWIDE EXPANSION, AND WHAT 2017 HAS IN STORE
FOR COMPUTING & DIGITAL MAKING EDUCATORS
Issue 1 | Spring Term 2017 | helloworld.cc
PAPERT'S LEGACY
What we can learn from the father of Logo, and constructionism's role today
LEARN NEW COMPSCI SKILLS
Start plugging knowledge gaps today
PROJECT QUANTUM
Do your pupils really understand? Quantum aims to help you ask the perfect question
LESSON PLANS AND TUTORIALS
Ideas to inspire and help you and your students
CREATE A MAKERSPACE
How to introduce a digital making space in your school
PLUS
IS COMPUTING BECOMING MORE EXCLUSIVE? • ITALY PUTS MAKERSPACES IN EVERY SCHOOL • GRADUATE FROM SCRATCH TO PYTHON • CODING INTERACTIVE FICTION • PIONEERS PROGRAMME LAUNCHES • YOUR DIGITAL MAKING QUESTIONS ANSWERED • 3D GRAPHICS WITH FREE SOFTWARE • INSIDE PICADEMY USA • THE IMPORTANCE OF PLAYFUL COMPUTING

Hello World is a new magazine from the Raspberry Pi Foundation that's free to all educators

NOW TRENDING

The stories we shared that flew around the world



SHARE YOUR POWER WITH BOINC

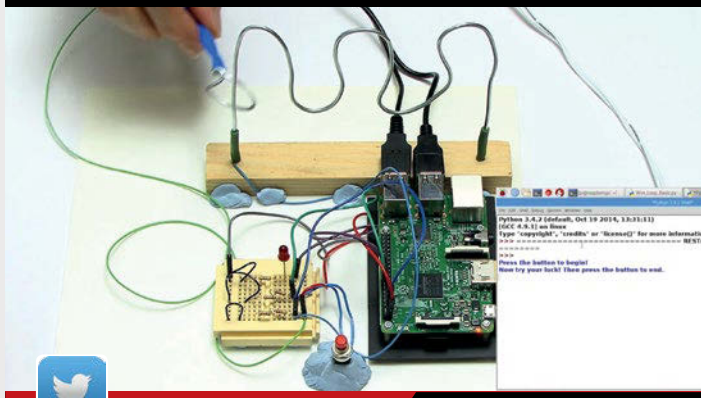
magpi.cc/2l2sQzl

Got a Raspberry Pi that idles away most of its day? Lend some of its CPU power to science.

PROGRAMMING A RASPBERRY PI

magpi.cc/2jyzU4V

Discover the joy and art of computer programming with your Raspberry Pi.



WIRE AND LOOP GAME

magpi.cc/2hNaep3

This Wire and Loop game is a great project for learning basic circuits, electronics, and Python coding.

ROBOTS AT THE SCIENCE MUSEUM

Above Baxter is the world's first two-armed robot designed to work together with people

New exhibition brings 100 robots under one roof

Over 100 robots are on display at a new exhibition at the Science Museum, London.

The exhibition is called 'Robots' and is set to last all summer.

It "explores humanity's 500-year quest to reimagine ourselves as machines," explains the museum.

Robots is open daily until 3 September 2017. Tickets are on sale now (**magpi.cc/2kLvyeO**).

Ian Blatchford, director of the Science Museum Group, says: "This stunning exhibition explores the fascinating question of why, rather than how, we build robots. To look through the eyes of those who built, commissioned or gazed upon these mesmerising mechanical creations over the past 500 years reveals so much about humanity's hopes, fears, dreams, and delusions."

Pepper is a humanoid companion created to communicate with humans through voice and touch and movement. This robot invites visitors to touch its hands and fist-bump while sharing a story using the screen on its chest.

Baxter, REEM Service Robot, and Amico are industrial robots with amazing precision, while robots like ROSA, Inhka, and RoboThespien are used to interact with humans.

Some of the robots are incredibly lifelike. Zeno R25 has incredible expression, while Nao will dance, blow kisses, and exercise.

Diving head-first into the uncanny valley is Animatronic baby. The baby breathes, sneezes, and moves its arms and legs. It's designed to make you feel strong emotions towards a robot.

Ben Russell, the lead curator of Robots, says: "Coming face to face with a mechanical human has always been a disconcerting experience. That sense of unease, of something you cannot quite put your finger on, goes to the heart of our long relationship with robots."

Below RoboThespian was the first full-sized humanoid robot to be commercially produced

Below right Rob's Open Source Android was built to move like a human

All images courtesy of the Science Museum

GOOGLE TOOLS

Tech giant asks makers for advice on AI tools

Google is going to arrive in style in 2017. The tech titan has exciting plans for the maker community.

It intends to make a range of smart tools available this year. Google's range of AI and machine learning technology could enable makers to build powerful projects.

To make this happen, Google needs help from the maker community. Raspberry Pi fans are the best makers around, and it's their ideas that will give the tech company direction.

Here's what Google has to say: "Hi, makers! Thank you for taking the time to take our survey. We at Google are interested in creating smart tools for makers and want to hear from you about what would be most helpful. As a thank-you, we will share our findings with the community so that you can learn more about makers around the world."

The company can produce some serious tools for the maker community, so make sure you have your say to get the tools you need.

Let Google know what you would like by filling out the survey: **magpi.cc/2l152MR**.

DEXTER INDUSTRIES

GoPiGo

20+ new projects for your Pi robot car.

BrickPi3

LEGO MINDSTORMS + RASPBERRY PI

www.dexterindustries.com

RETRO GAMING BLISS

Build upon Adafruit's amazing **Pi GRRL 2** using the brand new **Pi Zero W** to create the ultimate retro handheld console…

Portable gaming has been hugely popular ever since the Nintendo Game Boy was released in the late eighties. Building on the work of the Game & Watch LCD games, the Game Boy allowed you to take one machine wherever you went. It played a multitude of games via handy cartridges filled with code, including video game classics such as Tetris, The Legend of Zelda: Link's Awakening, and Pokémon.

Since then, handheld gaming and computers have evolved. Mobile phones have become a great source for providing quick hits, while Nintendo is going all out with its hybrid Switch system. On the computing side, processing power has advanced to such a degree that the Raspberry Pi is powerful enough to emulate several popular retro home consoles, while also being small enough to carry around.

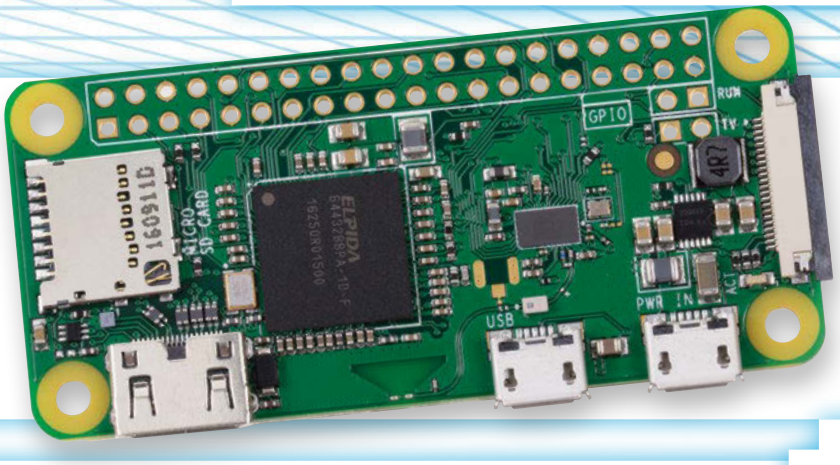
This is why the PiGRRL projects from Adafruit are popular: handheld, old-school consoles you can use on the go. There are many versions of them, based on everything from original models of the Raspberry Pi to the Pi Zero.

With the release of the Pi Zero W, these projects can go even further thanks to the built-in wireless LAN, and it also leaves more space for a bigger battery. And a bigger battery means longer play time. In this guide, we're going to show you how to take the PiGRRL 2 and do just this. Grab your work dungarees and let's-a go!

SUPER
MARIOWORLD™
©1990•1991 Nintendo

ASSEMBLE YOUR PARTS

Here's what you'll need to make the **Pi GRRL Zero W**



Pi Zero W

> magpi.cc/2l6zurq <

The key to this project is the Pi Zero W. The wireless-enabled version of the Pi Zero is as powerful as the Pi Zero, yet saves a load of space thanks to the radio chip included in it, so no WiFi dongle is needed. It runs at 1GHz, which makes it powerful enough to run emulators of many eighties and nineties consoles.

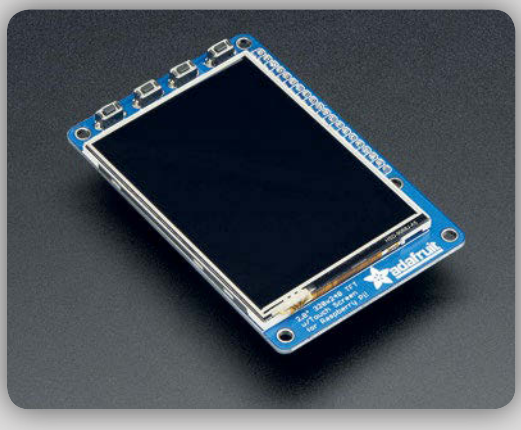
TOOLS FOR THE JOB

- Soldering iron
- Wire
- Heat shrink
- Glue
- Blu-Tack
- Wire strippers
- Hobby knife

3D-printed case

> magpi.cc/2kSgK1f <

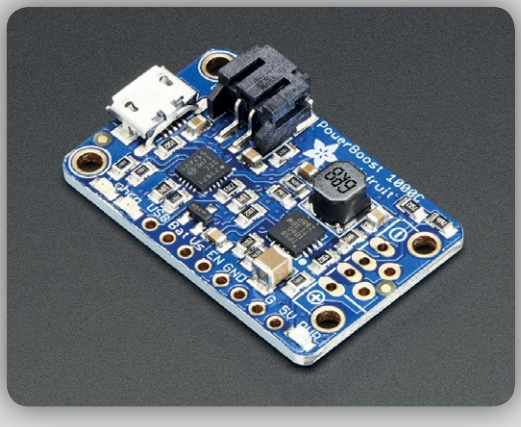
The PiGRRL 2 case repurposed for our needs. It has some spaces for USB and Ethernet on a B+/2/3 board, but the Pi Zero will require extensions if you want to make use of the gaps.



Adafruit PiTFT 2.8"

> magpi.cc/2lob5Ky <

This is actually a touchscreen, although we won't be making use of it in that way. It fits neatly in the case and provides four extra buttons to use when playing games. You can also assign system and UI shortcuts to the buttons.



PowerBoost 1000C

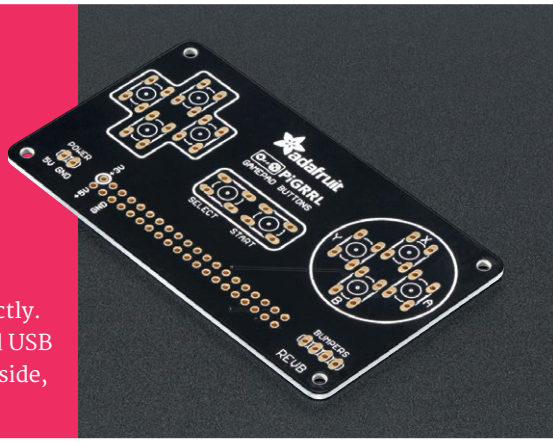
> magpi.cc/2lo5aFg <

This is one of the cool bits: we're going to use this PowerBoost to actually charge a battery within the handheld. With the low power draw of the Pi and advancements in modern battery tech, you'll get a lot out of one charge.

PiGRRL 2 controller board

> magpi.cc/2lohZzr <

This Custom Gamepad PCB is designed to fit the case perfectly. We did try to see if a standard USB SNES or NES PCB would fit inside, but they were far too big.



Slide switch

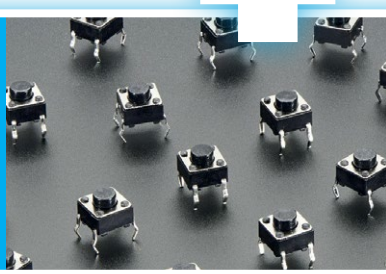
> magpi.cc/2lojfTb <

This switch allows you to turn the power on and off. It's best to do the software shutdown of RetroPie first before switching the power off, though.

Microswitches

> magpi.cc/2lo890i <

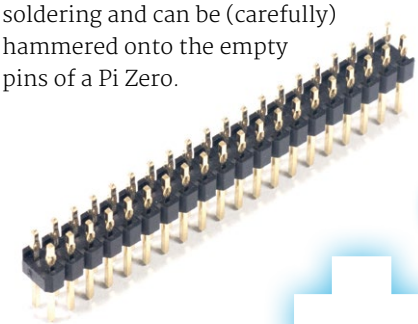
The beauty of proper retro gaming is tactile controls. You'll need ten 6mm switches, and a couple of 12mm buttons.

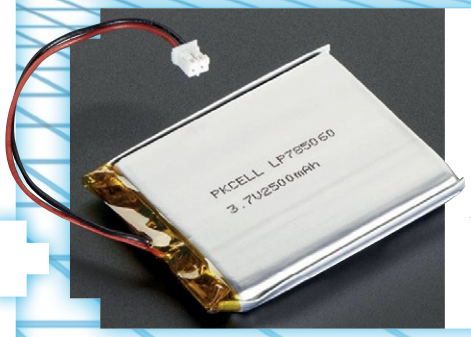


GPIO hammer headers

> magpi.cc/2lohN2U <

A wonderful innovation from Pimoroni, these GPIO headers require zero soldering and can be (carefully) hammered onto the empty pins of a Pi Zero.





2500mAH battery

> magpi.cc/2lQzVmr <

We squeezed the biggest battery into here that we possibly could. This way, it should last for hours and hours.

Bluetooth audio (optional)

There's no dedicated Audio Out on the Pi Zero W, but it does have Bluetooth. RetroPie doesn't support Bluetooth audio yet, but there are workarounds: see **magpi.cc/2oMboUz**

SCREWS

A selection of screws to attach the parts to the case. This includes 14× #4-40 and 6× #2-56 3/8 machine screws.

PRINT THE CASE

What you need to know about 3D-printing the **Pi GRRL 2** case

The proliferation and advancement of 3D printing has been a huge boon for the maker community, enabling you to create wonderful chassis and cases for your final products. The PiGRRL series has a number of cases built around the Raspberry Pi that allow for maximum efficiency in size, while also allowing for a fully operational handheld.

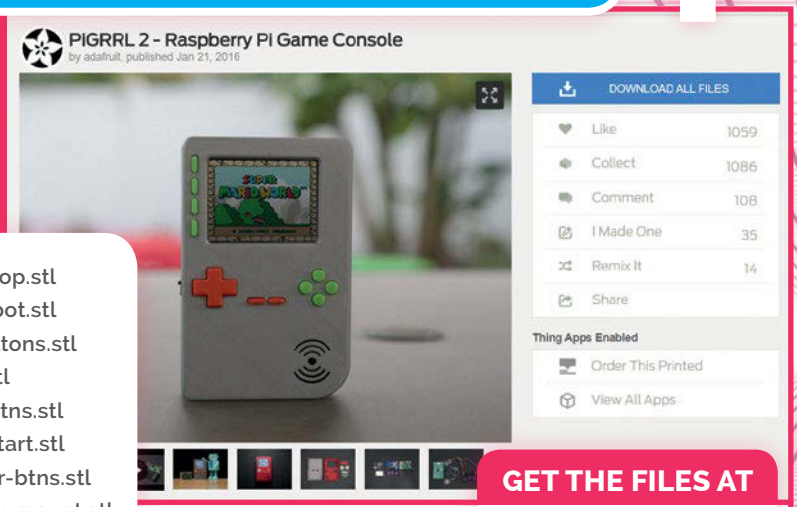
For this project, we're going to make use of the slightly larger PiGRRL 2 case for maximum comfort, and also so we can use the extra space to install a bigger battery into it. Here's how to make your own.

HOW TO PRINT YOUR 3D CASE

>STEP-01 GET THE FILES

The full PiGRRL 2 case files can be downloaded from **magpi.cc/2kS9K1f**, although there are more files here than what you actually need to print. The ones you'll need from the pack are:

- pigrrl2-top.stl
- pigrrl2-bot.stl
- pitft-buttons.stl
- dpad2.stl
- action-btns.stl
- pause-start.stl
- shoulder-btns.stl
- shoulder-mount.stl



GET THE FILES AT
magpi.cc/2kS9K1f

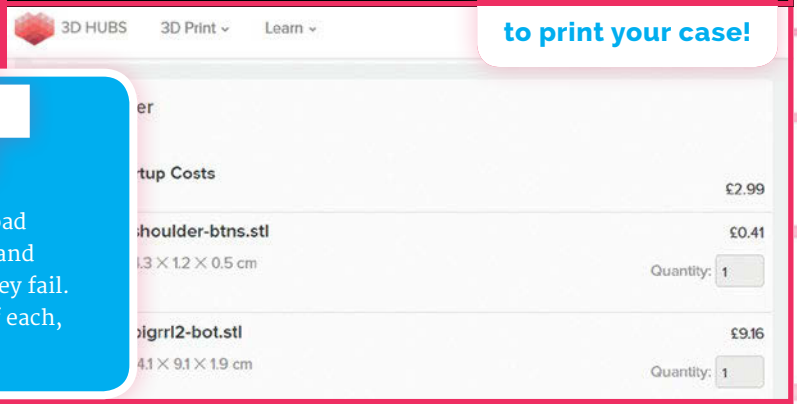
>STEP-02 FIND A 3D PRINTER

It can be tricky to find a good 3D printing service online, so unless you have access to a 3D printer, we highly recommend using **3DHubs.com**. It lists local 3D printing services, along with an estimated completion time and reviews. The files we downloaded also work with the service.

USE 3DHUBS.COM
to print your case!

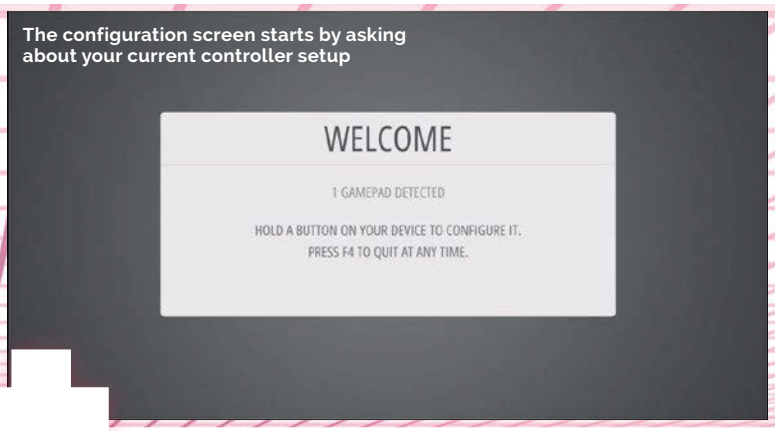
>STEP-03 UPLOAD FILES

Once you've selected your printer, you'll be asked to upload the files. Double-check you've selected the correct ones and make sure they upload properly – you'll get an error if they fail. Usually, trying again will work. You also only need one of each, and ABS or PLA are great materials to use for the parts.

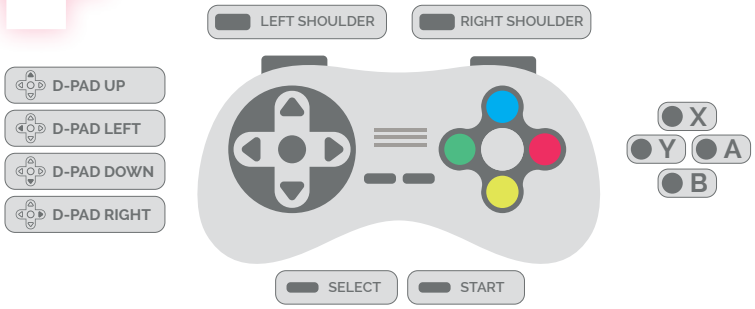


PREPARING YOUR PI

Get your **Pi Zero W** ready to be made into a retro gaming treasure



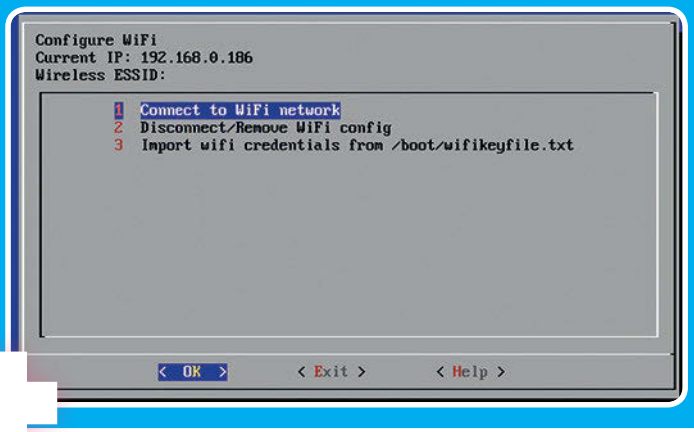
The configuration screen starts by asking about your current controller setup



When you come back to reconfigure your buttons, refer to this handy guide

SETTING UP WIRELESS

Wireless connectivity on the new Pi Zero W is great, but the configuration method for wireless LAN on the RetroPie is very basic. You'll need to know the name of your wireless network (SSID) along with the password, as it won't be able to search for available networks. You can also import details by loading a .txt file onto the boot partition.



For this project, we're going to be using the excellent RetroPie to power our emulation software. You'll first need to download the image for RetroPie from its website here: **magpi.cc/25UDXzh**. Write it to an SD card and pop it into your Pi Zero W. Get that hooked up to a monitor along with a keyboard, and we can get it ready.

You'll first need to map some buttons – as the PCB controller isn't hooked up yet, we'll have to quickly use the keyboard for the initial setup. Make sure the directional keys, Start, Select, A, and B are assigned a key and just hold down the space bar to skip anything else. Once that's done, connect to the wireless using the info in the 'Setting up wireless' boxout.

To get our final build working, we need to make sure to install support for the PiTFT, as it's not supported natively. SSH into the Pi Zero W at **retropie.local** or press **F4** to enter the command line on RetroPie, and enter the following:

```
$ cd
$ curl -O https://raw.githubusercontent.
com/adafruit/Raspberry-Pi-Installer-
Scripts/master/pitft-fbcp.sh
$ sudo bash pitft-fbcp.sh
```

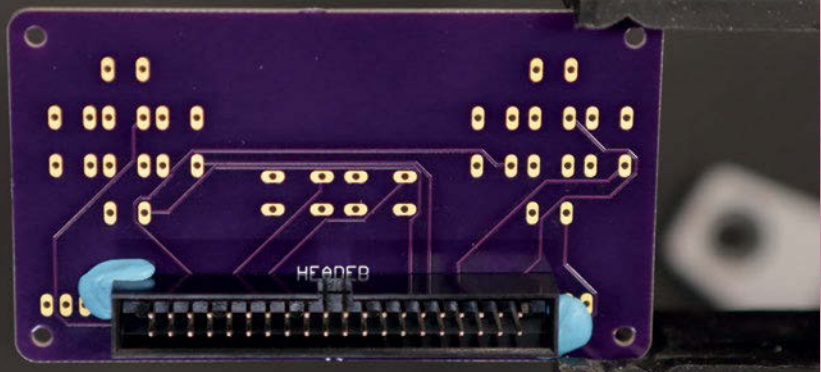
Select PiGRRL 2 and don't reboot. Now we need to add support for the custom buttons. Back in the command line, use:

```
$ cd
$ curl -O https://raw.githubusercontent.
com/adafruit/Raspberry-Pi-Installer-
Scripts/master/retrogame.sh
$ sudo bash retrogame.sh
```

Select PiGRRL 2 again and then reboot the system. Once we've put all the parts together, and before it's assembled in the case, we'll need to configure the controls for the buttons we've made and added to the project. Press the button assigned to Start and select Configure Input, and then go through the configuration process again.

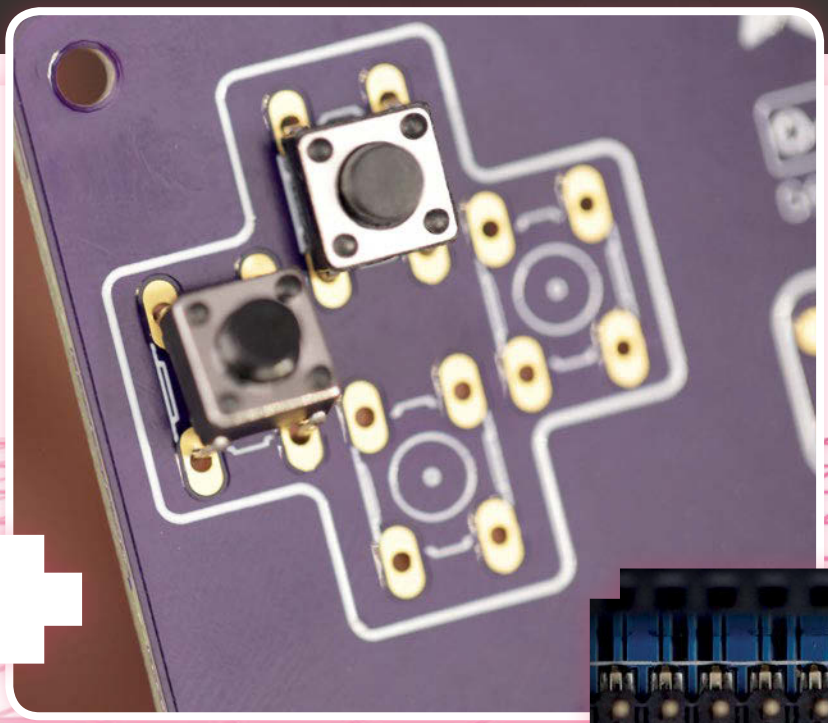
BUILD THE SYSTEM

Follow along and build your **retro handheld**



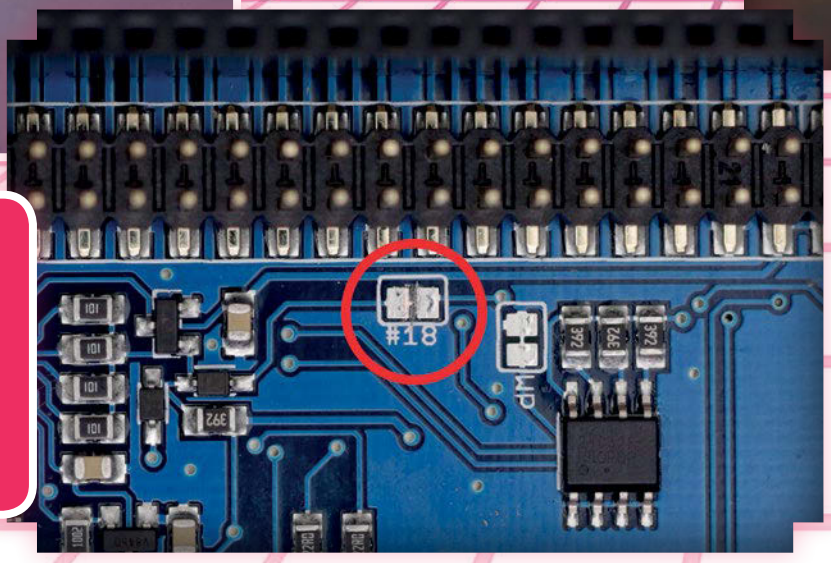
>STEP-01
PREPARE THE GAMEPAD BOARD

Our first job is to solder the header pins onto the Gamepad board. You can keep it all in place with a little Blu-Tack before soldering it on. Make sure you're soldering the header onto the correct side of the board.



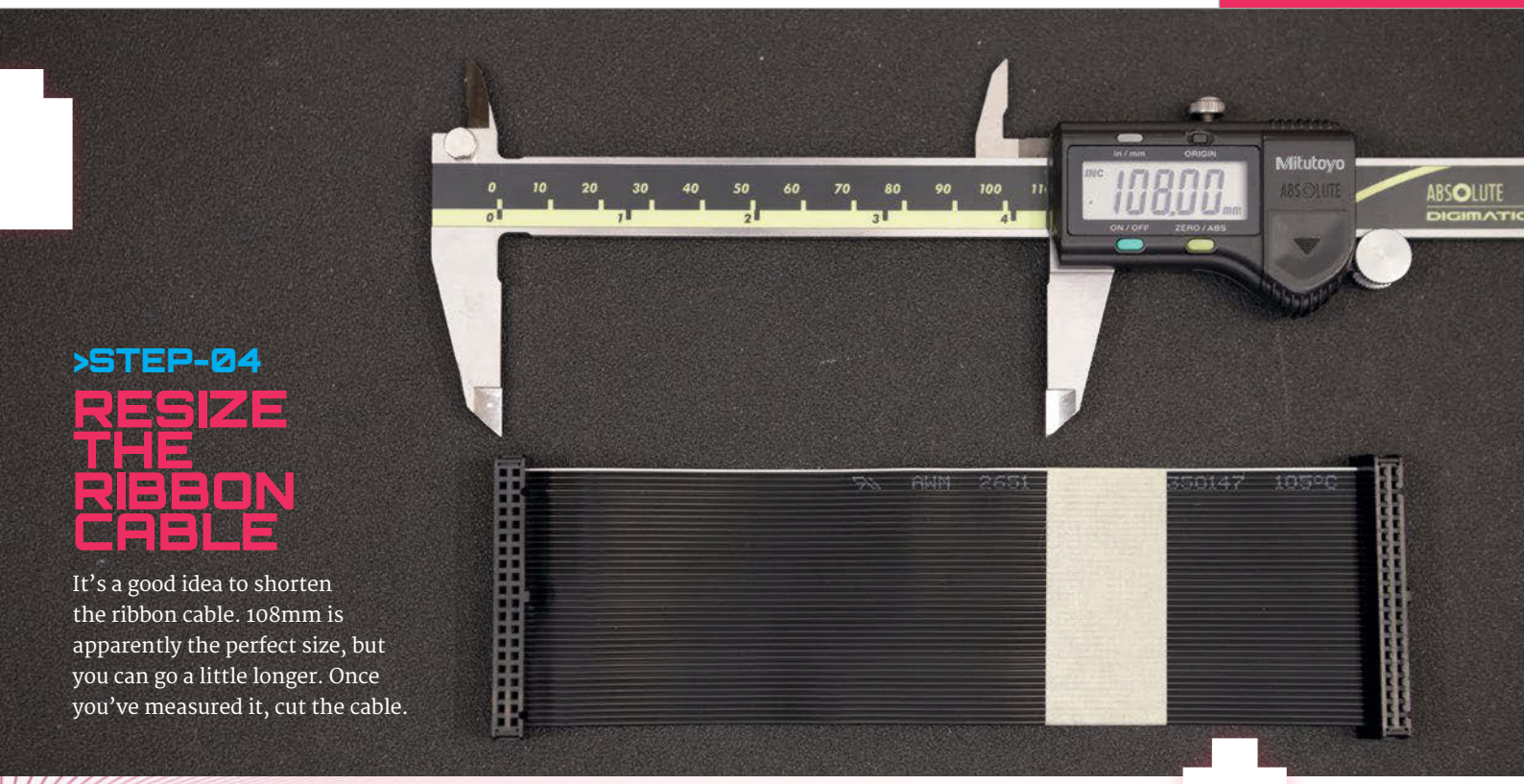
>STEP-02
ADD THE BUTTONS

Now it's time to carefully solder the ten 6mm microswitches to the front of the board. Some helping hands would be good here.



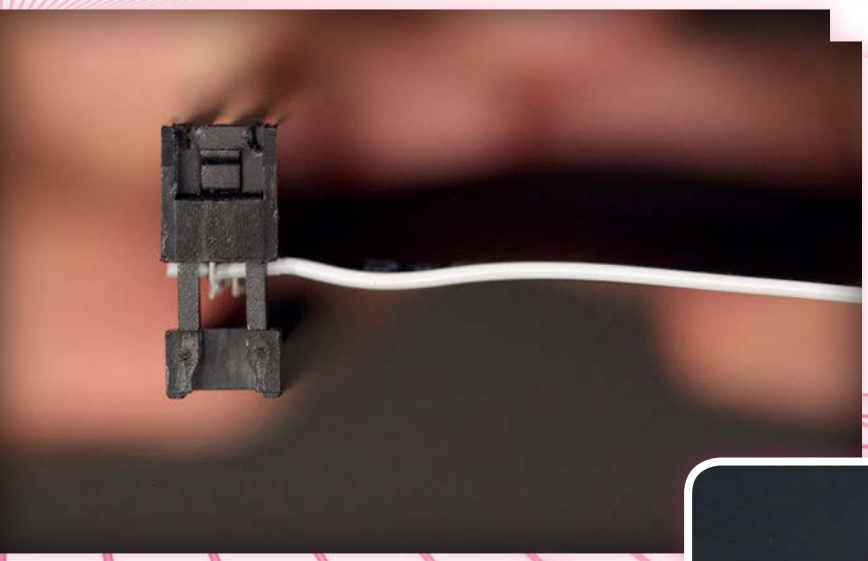
>STEP-03
TURN ON THE BACKLIGHT

By default, the PiTFT doesn't have its backlight turned on. You need to take a craft knife and cut the circuit between the blocks in the #18 box that you can see circled in the picture.



>STEP-04
RESIZE THE RIBBON CABLE

It's a good idea to shorten the ribbon cable. 108mm is apparently the perfect size, but you can go a little longer. Once you've measured it, cut the cable.

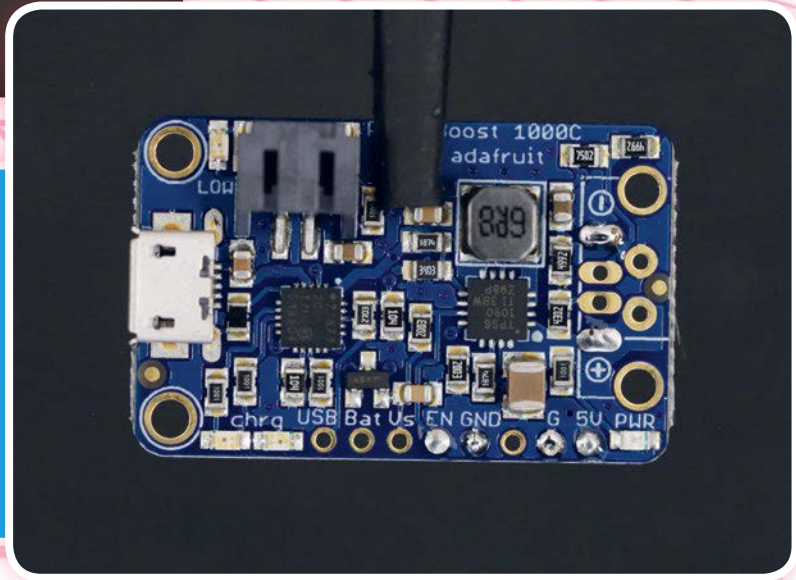


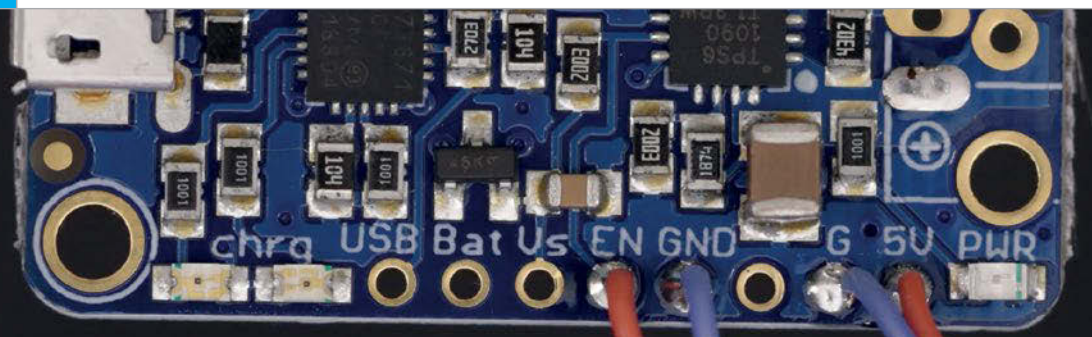
>STEP-05
ASSEMBLE THE CABLE

Using something like a pen or pin, you'll need to push in the clip that holds the connector in place on the part you're discarding. Very carefully remove the cable and install it at the end of your newly trimmed cable.

>STEP-06
PREPARE FOR POWER

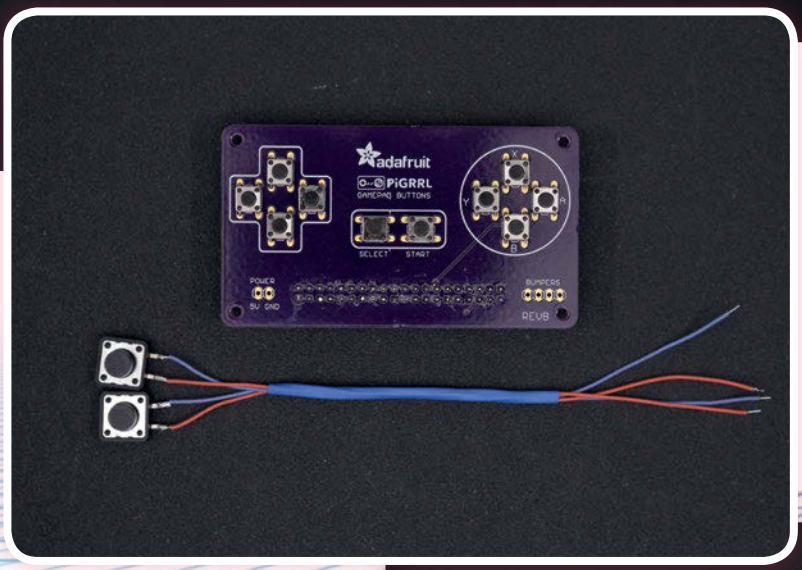
To make all our soldering easier, we'll dab some solder onto the spots where we need it for the moment. On the PowerBoost 1000C, add some solder to the positive and negative pins, and the EN and GND pins. Cut one of the legs off the power switch and put some solder on the other two.





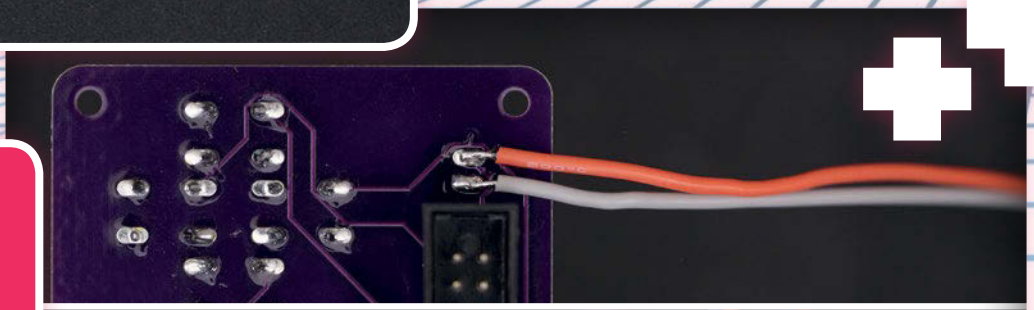
>STEP-07
SOLDER ON THE SWITCH

Make sure the power switch will fit in the hole for it in the case – it's on the side of the bottom part. You may need to file away the plastic a bit. Once that's done, trim two short bits of wire to about 7cm long and solder one to each leg. Solder those to EN and GND – it doesn't matter which way around they go.



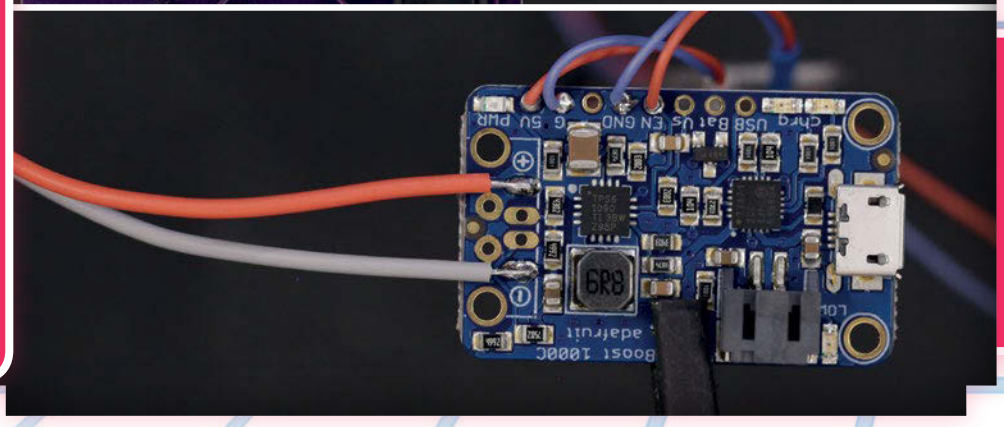
>STEP-08
SHOULDER BUTTONS

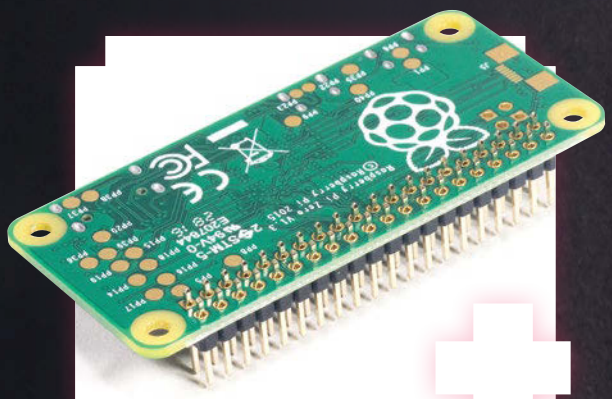
The shoulder buttons (the 12mm ones) need to be attached to the controller board much like the switch was connected to the power board. Clip two legs off each and use a pair of pliers to flatten the remaining two. Solder wires to each pin (about 14cm long) and then solder the other end to the bumper pins on the controller board. Again, polarity doesn't matter, but keep each one as a pair in the row.



>STEP-09
WIRE IT ALL UP

Now we can combine the power with the controller board, which will allow us to provide power to the whole system. Solder two wires (about 14cm long) to the underside of the 5V and GND pins on the controller board. The 5V wire should then be soldered to the positive of the PowerBoost board, with the GND to the negative.





>STEP-10
HAMMER THE HEADS

It's now a good idea to add the GPIO headers to the Pi Zero W. Gently hammer them in until they're secure, and you're done.

TEST IT ALL OUT!

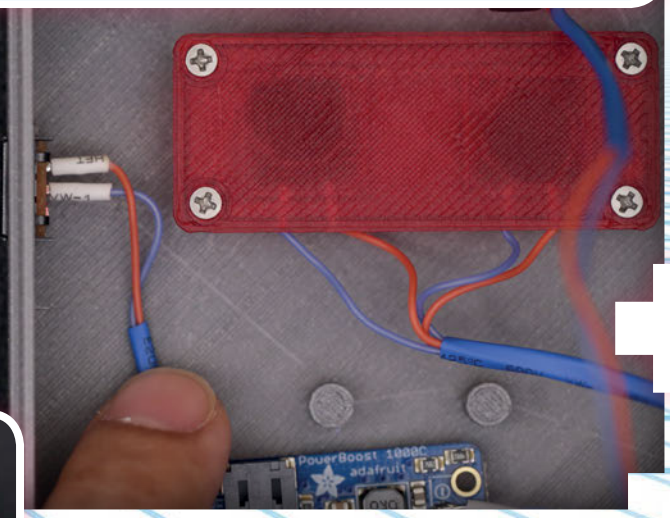
At this point you can test the entire system, otherwise we'll move onto final construction...

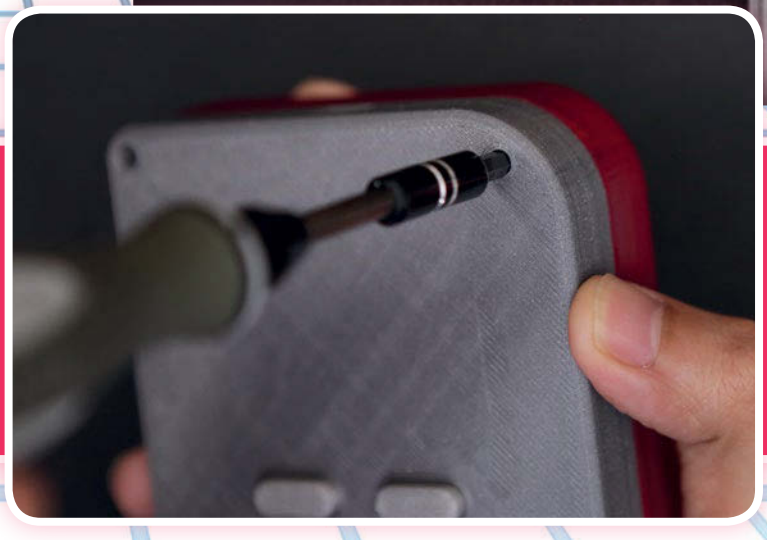
>STEP-11
BEGIN CONSTRUCTION!

Now the easy bit. Take the top of the case and insert all the button 3D prints. Once that's done, insert the screen and screw it in, followed by the controller board. Finish up the top bit by connecting the two with the ribbon cable, and then insert the Pi Zero into the header on the PiTFT.

>STEP-12
BACK PLATE

You'll now need to mount the back buttons and the PowerBoost. The rear buttons have a plate that keeps them in place, and Adafruit suggests using a little Blu-Tack to stick the switches in position. You'll attach the power switch and then finally screw on the PowerBoost. Be careful with the wires that you've soldered on.





>STEP-13
CLOSE IT UP

Now you can finally close it up! Insert the battery and tape it down if need be, before screwing it shut. Make sure all the cables are safely inside the case before tightening it, though!

WANT TO KNOW MORE?

Prefer to use a full-sized Pi? Check out the original PiGRRL 2 guide **magpi.cc/2l0ALzQ**

USING A WIRELESS PIGRRL

Here's some of the amazing advantages of having the **PiGRRL** powered by **Pi Zero W**

CONNECT VIA SSH

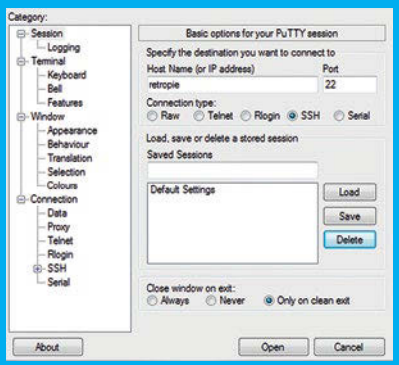
Physically connecting to the Pi Zero inside is a massive hassle once the case is screwed together. With the Pi Zero W connected to your home network, though, it's easy to connect to it remotely from another computer using SSH.

On a Mac or Linux machine (which includes another Raspberry Pi!), you can simply open the terminal or command line and enter the following to connect:

```
ssh pi@retropie
```

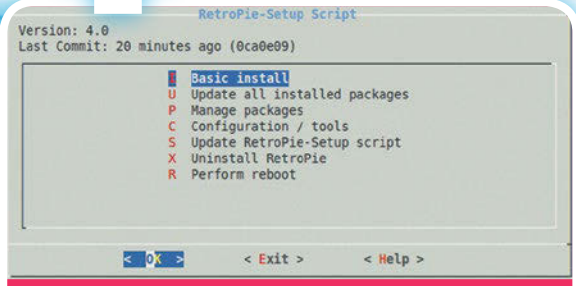
It will prompt you for a password, which is **raspberry**. The user name in this instance is **pi**, with **retropie** being the default name for the system on the network.

For Windows machines, you'll have to connect using an SSH application like PuTTY (**magpi.cc/2lBHCRm**). Once it's installed, you need to set the host name to **retropie**, the port to 22, and then click Open. You'll need to put in **pi** as the user name and **raspberry** as the password.



You can ignore a lot of the PuTTY interface when just connecting to the handheld

Once inside, you can control many aspects of the system via the command line. If you've used the terminal in Raspbian, you'll know how it works: sudo reboot, ls, cd, etc.



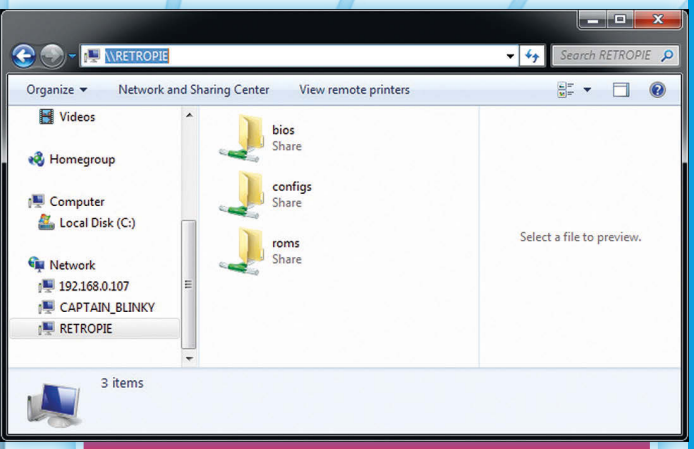
UPDATING RETROPIE

You can update RetroPie from the handheld itself or via SSH in the command line. The keyboard might be better suited for this, so if you're by your computer it wouldn't hurt to use it instead.

From the handheld, you need to go to the RetroPie menu in EmulationStation and activate the setup script. From the terminal (if you've SSHed in), you can use the following command:

```
sudo ~/RetroPie-Setup/retropie_setup.sh
```

From here, look for the 'Update All Installed Packages' option. There are many options here that you can select from, including managing the individual packages in case you want to remove or add any. To update, you can select the option 'Update all installed packages' (which will also update the **RetroPie-Setup** script as well) or you can go to 'Manage packages' and update the packages individually. This could be useful if any packages have some problems updating, or if you want to do the essential updates before running out the house.



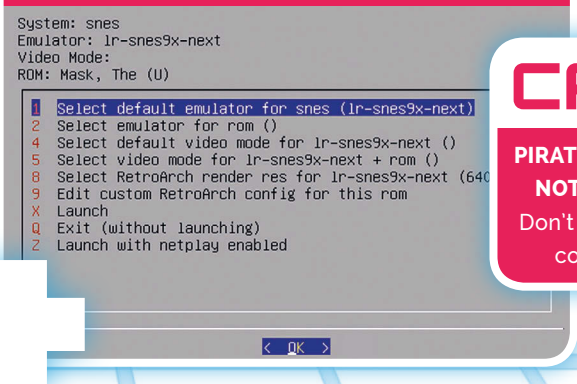
UPLOADING FILES

If you want to upload ROMs to the handheld, you can do so with the Pi Zero W's wireless connection. Otherwise, you'd have to manually load them onto the SD card, which would require dismantling the console to get to the Pi Zero – not particularly easy or practical to do. Luckily, RetroPie includes Samba and SFTP, which allow you to transfer the files over the network.

For Samba it's nice and easy: when your handheld is connected to your network, you can go to your main computer and find it on the network shares as \\RETROPIE. Here you can upload any necessary extra files to the handheld with minimal hassle.

For SFTP you'll have to make use of special software. For Windows, the RetroPie team recommend WinSCP (**magpi.cc/2lCwRhz**); for Mac, you can try Cyberduck (**magpi.cc/2lCwjs9**).

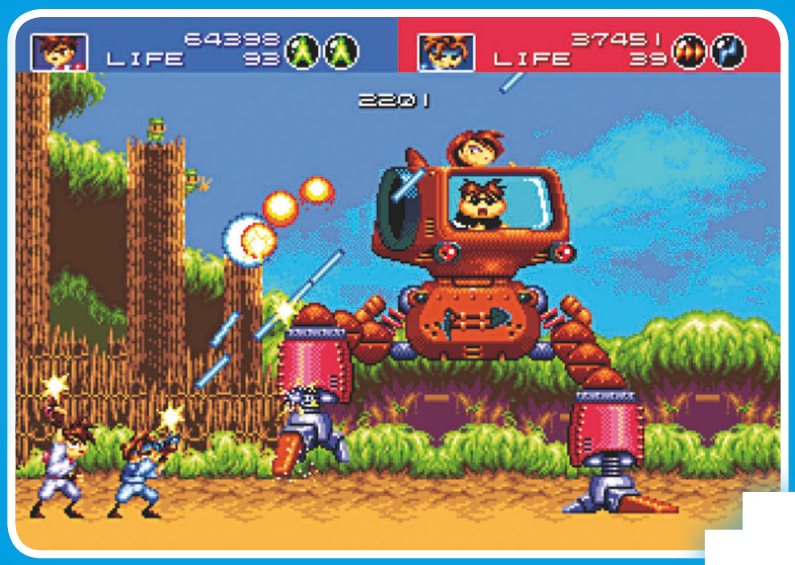
Once booted up, you can use the same SSH settings as we used for PuTTY. You can then drop the files into the corresponding folder in the **roms** directory.


System: snes
Emulator: lr-snes9x-next
Video Mode:
ROM: Mask, The (U)
1 Select default emulator for snes (lr-snes9x-next)
2 Select emulator for rom ()
4 Select default video mode for lr-snes9x-next ()
5 Select video mode for lr-snes9x-next + rom ()
8 Select RetroArch render res for lr-snes9x-next (640
9 Edit custom RetroArch config for this rom
X Launch
Q Exit (without launching)
Z Launch with netplay enabled
< OK >

CAUTION!

PIRATING VIDEOGAMES IS BAD, NOT TO MENTION ILLEGAL!

Don't use illegally downloaded content on your PiGRRL.

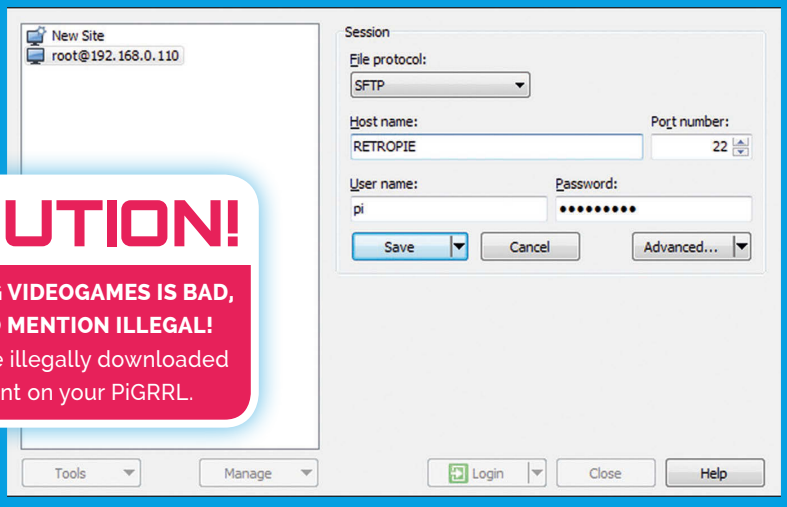


PLAYING ONLINE

Otherwise known as netplay in emulation circles, this allows you to play multiplayer games with friends, even if they're on the other side of the world! Let's see the SNES do that! Not every emulator supports it and if it does, you need to follow three extra rules: both parties need to be running the same version of RetroArch, both must be running the same emulator, and both need to be running the same ROM.

You then need to configure netplay from the setup script. If you're hosting, change your Netplay Mode to host, make the host IP your IP address, and pick a nickname. The client (other player) needs to go to the same menu, change Netplay Mode to client, change the host IP to the other handheld, and pick your nickname. You may need to open up a specific TCP/UDP port on the host's router, which you then need to set as the same on both systems.

Now both of you need to open the same ROM using the 'jso' key (which should be X on a SNES layout) and select 'Launch with netplay enabled'. If it's been set up correctly, you'll connect!

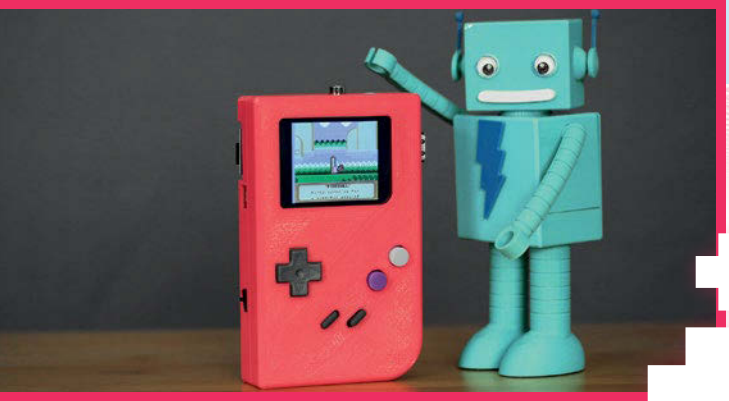


MEET THE RUIZ BROTHERS

We talk to the original creators behind the **Pi GRRL** and many other amazing **Adafruit** projects

PIGRRL HISTORY

How the **PiGRRL** project has evolved



PIGRRL

The original PiGRRL celebrated the 25th anniversary of the Game Boy and used the original Raspberry Pi Model B to power it. It's a lot bigger than the version we're building, although it more accurately matches the size of the original Game Boy.

PIGRRL 2

This should look familiar – this is the version we've based ours on! It's an upgrade over the PiGRRL as it uses a lot more custom components, including a custom PCB for the controls instead of a repurposed SNES board. You could easily switch a Pi 3 in there if you wanted a bit more power.

Two of the superstars of the maker scene are Noé Ruiz and Pedro Ruiz, otherwise known as the Ruiz Brothers. They've done many amazing projects for Adafruit, including a lot of 3D printing and wearables, which always go down well with the community. So it's surprising to hear they've only been in the maker scene for about five years.

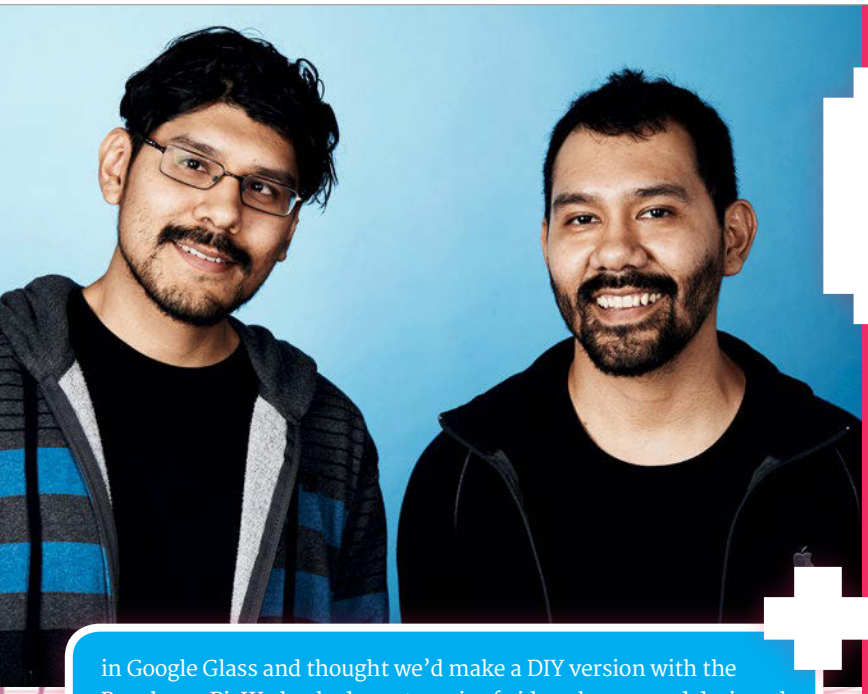
"My brother and I purchased our first 3D printer in 2012 and quickly started using it in our work," Noé tells us. "While looking for a way to integrate lighting into our 3D-printed designs, we discovered Adafruit and the Arduino platform. We built some projects using their parts and came up with some unique ideas. We went on Adafruit's weekly live show-and-tell show, and the rest is history."

What started off as a load of cool hacks that added LEDs to existing products or enabled you to create great light-up projects quickly evolved into doing more. Part of this was to do with the introduction of the Raspberry Pi.

"Our first project with the Raspberry Pi was the DIY Wearable Pi with Near-Eye Video Glasses," explains Noé. "We were interested

PIGRRL ZERO

The latest version of the PiGRRL is a tiny device, reminiscent of the GBA or the Game Boy Micro. It uses a Pi Zero and a series of other small components, all squeezed into a tiny little 3D-printed case. The Pi Zero is still powerful enough to run a lot of emulators, though.



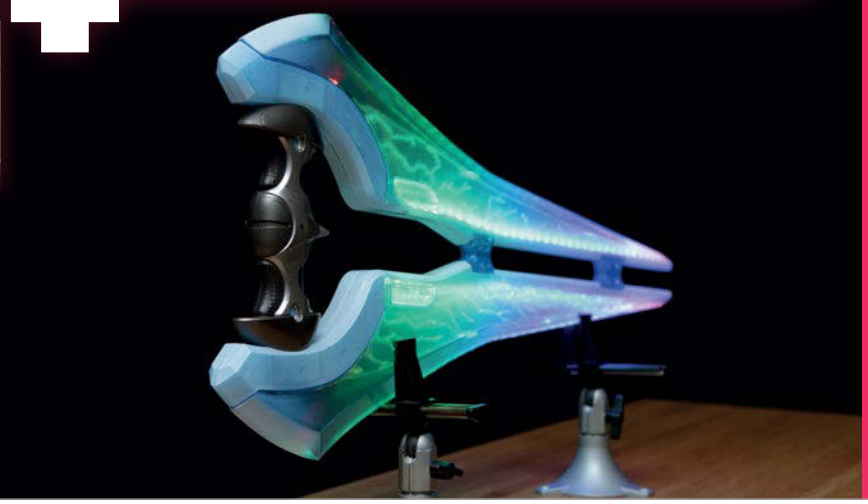
in Google Glass and thought we'd make a DIY version with the Raspberry Pi. We hacked apart a pair of video glasses, and designed a custom 3D-printed housing for the display and driver. It was a fun experiment and this is how we learned about the Raspberry Pi."

Things escalated even further when the brothers made the original PiGRRL. Originally an idea from Limor Fried, founder of Adafruit and 'Ladyada' herself, the idea was to improve upon her earlier Game Grrl project but this time use a custom 3D-printed enclosure. It was their biggest project to date, so they were extremely happy to see that it had such a positive reaction.

The PiGRRL projects have since become the Brothers' favourites to work on, according to Noé.

"I think it's become a classic Raspberry Pi project because it looks like an iconic device that offers lots of playtime. People love to play games, and being able to build your own gaming console is super-rewarding. Every year we create a new version with better hardware, and change the form factor to try different designs. So many folks have built one and it's really awesome to see parents building them with their kids."

What's in the future for the PiGRRL and the Ruiz Brothers? Well, PiGRRL 3 is happening this year with a "bigger screen and better audio", and should be a much quicker build. Watch this space!



MORE FROM THE RUIZ BROTHERS

Other amazing things you can find by **Noé** and **Pedro**

RASPBERRY PI POKÉMON FINDER

> magpi.cc/2lKXpcA <

Based on the Lure Modules found in Pokémon GO, this project caused a little bit a stir at the time, as it used some APIs that people possibly shouldn't have had access to. It would tell you if there were any Pokémon in your immediate area, and even display a coloured light for how rare the Pokémon was.

NEOPIXEL YOYO

> magpi.cc/2lKRd4c <

A simple playground toy turned awesome with the use of some NeoPixel LEDs. The yoyo itself is also 3D-printed, allowing for custom parts so that you can fit the electronics inside. It even has a USB charging port. Check out the link for some cool GIFs of the yoyo in action.

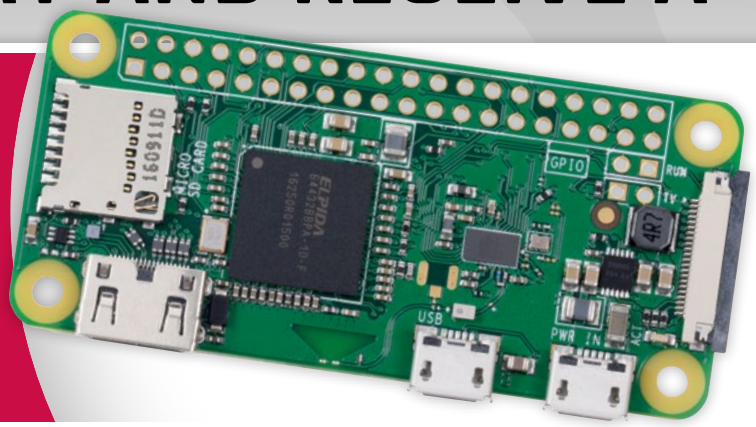
HALO ENERGY SWORD

> magpi.cc/2lKKSWC <

This one isn't a completely custom build – instead, it's an upgrade/customisation of a pre-existing licensed Halo Energy Sword toy by Mattel. The Ruiz Brothers took the already pretty cool design and added a ton of NeoPixels to make it pulse with energy, similar to how the sword looks in the Halo games.

SUBSCRIBE TODAY AND RECEIVE A

FREE PI ZERO W



PLUS OFFICIAL PI ZERO CASE WITH 3 COVERS

AND FREE CAMERA MODULE CONNECTOR AND USB / HDMI CONVERTER CABLES

Subscribe in print for 12 months today and receive:

- A free Pi Zero W (the latest model)
- Free Pi Zero W case with 3 covers
- Free Camera Module connector
- Free USB and HDMI converter cables

Other benefits:

- Save up to 25% on the price
- Free delivery to your door
- Exclusive Pi offers and discounts
- Get every issue first (before stores)

NEW RASPBERRY PI ZERO W OUT NOW!
The MagPi
The official Raspberry Pi magazine
Issue 55 March 2017
BUILD A DIGITAL WHOOPEE CUSHION
Have a laugh with physical computing
MAKE AN ALEXA ROBOT
Take voice control with Amazon's Alexa
RETRO GAMING BLISS
Build this handheld gaming masterpiece with the brand new Raspberry Pi Zero W
BACK UP YOUR RASPBERRY PI
Keep your files safe
ADVANCED GPIO CONTROL
Improve your projects
Also inside:
THE TINY PI-POWERED SMARTPHONE
UPGRADE YOUR TV WITH A PI HTPC
3D MODELLING ON RASPBERRY PI
KODI 17 REVIEWED AND RATED
THE ONLY MONTHLY MAGAZINE WRITTEN BY AND FOR THE PI COMMUNITY
THE OFFICIAL RASPBERRY PI MAGAZINE
MAKE A NIGHT-VISION NATURE CAMERA
BUILD A MAGIC MIRROR
OBJECT-ORIENTED PROGRAMMING
FREE DEBIAN + PIXEL LIVE BOOTING DVD
BEGINNER'S GUIDE TO CODING

Pricing

Get six issues:

£30 (UK)

£45 (EU)

$69 (USA)

£50 (Rest of World)

Subscribe for a year:

£55 (UK)

£80 (EU)

$129 (USA)

£90 (Rest of World)

FREE PI ZERO W!

Get three issues:

£12.99 (UK) (Direct Debit)

$37.50 (US) (quarterly)

How to subscribe:

- magpi.cc/Subs-2 (UK / ROW)
- imsnews.com/magpi (USA)
- Call +44(0)1202 586848 (UK/ROW)
- Call 800 428 3003 (USA)

Available on the App Store

GET IT ON Google Play

SUBSCRIPTION FORM

YES! I'd like to subscribe to The MagPi magazine and save money

This subscription is: ☐ For me ☐ A gift for someone*

Mag#55

YOUR DETAILS Mr ☐ Mrs ☐ Miss ☐ Ms ☐

First name ..... Surname .....

Address .....

.....

Postcode ..... Email .....

Daytime phone ..... Mobile .....

**If giving* The MagPi *as a gift, please complete both your own details (above) and the recipient's (below).*

GIFT RECIPIENT'S DETAILS ONLY Mr ☐ Mrs ☐ Miss ☐ Ms ☐

First name ..... Surname .....

Address .....

Postcode ..... Email .....

PAYMENT OPTIONS

1 DIRECT DEBIT PAYMENT £12.99 every 3 issues (UK only)

Instruction to your bank or building society to pay by Direct Debit

DIRECT Debit

Please fill in the form and send to:
The MagPi, Select Publisher Services Ltd,
PO Box 6337, Bournemouth BH1 9EH

Service user number 8 3 8 7 7 3

Name and full postal address of your bank or building society:

To: The Manager Bank/building society .....

Address .....

.....

..... Postcode .....

Name(s) of account holder(s) .....

Branch sort code ☐☐☐☐☐☐ Account number ☐☐☐☐☐☐☐☐

Reference ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ (Official use only)

Instruction to your bank or building society

Please pay Select Publisher Services Ltd Direct Debits from the account detailed in this instruction subject to the safeguards assured by the Direct Debit Guarantee. I understand that this instruction may remain with Select Publisher Services Ltd and, if so, details will be passed electronically to my bank/building society.

Signature ..... Date ☐☐/☐☐/☐☐

Banks and building societies may not accept Direct Debit instructions for some types of account.

SUBSCRIPTION PRICING WHEN PAYING BY CHEQUE OR CREDIT/DEBIT CARD

6 ISSUES ☐ UK £30 ☐ Europe £45 ☐ Rest of World £50

12 ISSUES ☐ UK £55 ☐ Europe £80 ☐ Rest of World £90

2 CHEQUE

I enclose a cheque for ..... (made payable to Select Publisher Services Ltd)

3 CREDIT/DEBIT CARD ☐ Visa ☐ MasterCard ☐ Maestro ☐ Switch

Card number ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Expiry date ☐☐ ☐☐ Valid from ☐☐ ☐☐ (if shown)

Issue number ☐☐ (if shown) Security number ☐☐☐
(last 3 digits on the back of the card)

Signature ..... Date ☐☐/☐☐/☐☐

I would like my subscription to begin from issue ..... (month + year)

RETURN THIS FORM TO:

MagPi Magazine Subscriptions, Select Publisher Services Ltd, PO Box 6337, Bournemouth BH1 9EH

☐ Please tick this box if you DO NOT want to receive any other information from Select Publisher Services Ltd.

☐ Please tick this box if you DO NOT want to receive any other information from other companies.

☐ Please tick this box if you DO NOT want to subscribe to The MagPi newsletter.

ARSENIJS PIČUGINS

Arsenijs is an 18-year-old student from Riga, Latvia. He's a programmer and hardware repair person who works closely with Linux.
crimier.wordpress.com

ZEROPHONE

A SIM800 GPS module and antenna are used to connect to the mobile phone network
A small LCD screen displays the operating system (based on Raspbian Linux)
Buttons are connected to the PCB, and an ATMega328P module is used to convert button pushes into phone commands

Quick Facts

- The display resolution is 128×64
- Currently, it costs around $50 in parts
- It connects to a 2G network (a 3G model is planned)
- The OS is based on Raspbian Linux
- There are plans to crowdfund the project as a kit

Turning a Pi Zero into a mobile phone

ZeroPhone is an open-source project to create a mobile phone kit for the Pi Zero.

Owners of ZeroPhone will be able to make calls and send text messages. It will also have basic apps such as an alarm clock, calendar, calculator, web browser, and music player.

Behind the project is Arsenijs Pičugins, an 18-year-old student from Riga, Latvia. He now has a team of people helping him.

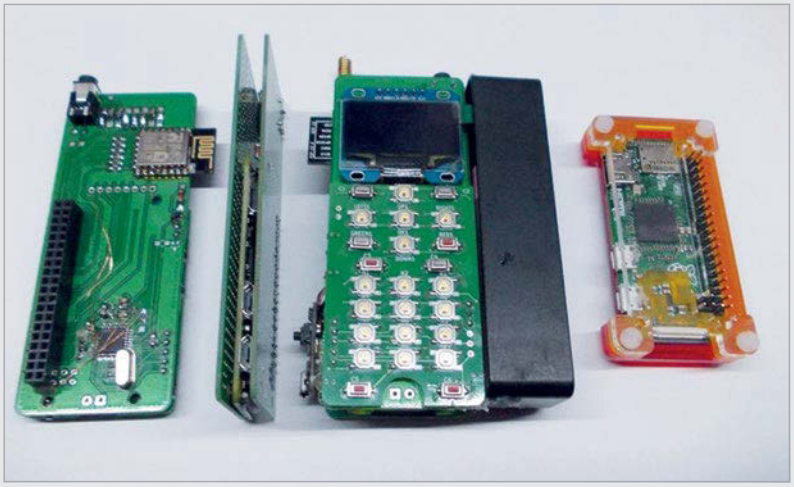
"It's open-source, Linux-powered, and you can make one yourself for $50," he says.

Arsenijs has been studying mobile phones for a couple of years. He hopes to sell a commercial kit, while keeping it all open-source for people to build themselves.

"The phone hardware is at 'alpha' stage," he tells us. Arsenijs has designed the first revision of PCBs (printed circuit boards) and is developing the software.

"The next revision will be ready-to-assemble," reveals Arsenijs. Raspbian is being customised to run on the ZeroPhone. "It can be tailored to suit our purpose very easily," explains Arsenijs.

"I've currently got a framework I'm using for headless Pi configuration," he continues. Python is being used to create a user interface (UI). An



Above The components and software are under development, and a kit will be made available when it's ready

earlier project called pyLCI (**magpi.cc/2kjv20A**) forms the base of the interface.

Developed by Arsenijs in 2016, pyLCI is a Linux control interface. It enables you to control and configure your Pi with SSH, networking, or using an HDMI monitor.

"I developed it because that's what I needed to improve my Raspberry Pi tinkering and development workflow," recalls Arsenijs.

"Developing a good UI is one of the primary goals of this project," Arsenijs adds.

"That's what I'm focused on, and that means extensively testing the hardware I've got now, to find everything that has to be fixed."

Once v1.0 is ready, he will publish the design files and instructions, so people will be able to start assembling their phones if they wish. He'll then prepare a crowdfunding campaign to fund a manufacturing run and software development.

"It's the only phone you can assemble by getting all the parts yourself," claims Arsenijs. It doesn't use any rare parts or fine-pitch soldering. "You can even breadboard this phone if you're dedicated enough.

> "I developed it because that's what I needed to improve my Raspberry Pi tinkering"

Because Linux is the base, it can support a range of programming languages. Arsenijs expects "original ideas" for apps. He will be working on productivity and healthcare apps.

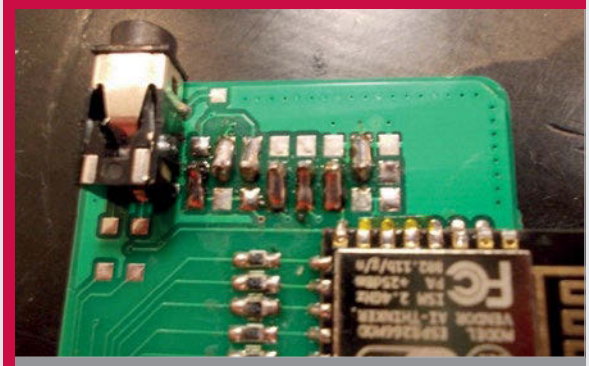
The open-source software approach is "likely to be secure and not privacy-invading," says Arsenijs. You can disable features you don't like, and it doesn't depend on any cloud services. You will be able to install firewall, I2P, and TOR nodes, as well as secure messenger services.

Arsenijs's current goal is to make the v1.0 PCB revision.

"For me, this project is a huge undertaking that, if done right, will influence the way we perceive mobile phones," remarks Arsenijs. He explains that it will help solve privacy and security concerns, as well as allow us to develop better open-source GSM technology. It will also help power mobile phone hacks.

"Besides, this is an open-source Linux-powered phone you can independently assemble for less than $50," observes Arsenijs. "That's pretty cool."

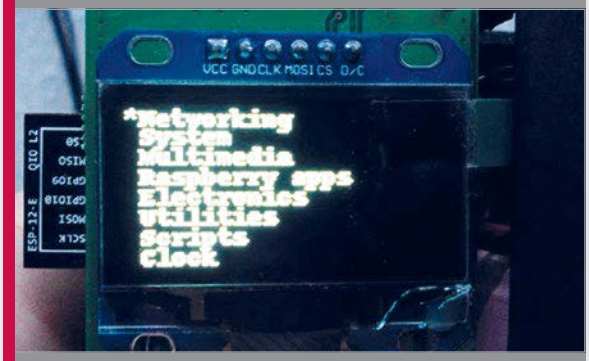
DESIGNING A PI ZERO PHONE KIT



>STEP-01

SIM module

A SIM800 module is used to connect to the GSM network. AT (Attention) commands, from the Hayes command set (**magpi.cc/2jvCYyO**), are used to control the modem connection to the modem network.



>STEP-02

LCD and battery

A small LCD screen is used to display the user interface. A battery with 18650 cells is used to power the Pi Zero, display, and SIM800 module.

>STEP-03

Controls

Two-layer PCBs (two 4×10cm and one 4×6cm) are used. Regular buttons are placed on the PCB to form a standard keypad. It's all hooked up to an ATMega328P that's used to read the button pushes.

MIKE HAMENDE

Mike Hamende hails from Manteno, Illinois. With a passion for making, his projects continuously gain him online recognition.
imgur.com/a/iOkCI

TRACKING TELESCOPE MOUNT

For Mike Hamende, making something is far more exciting and rewarding than logging into an online store, as his Pi-powered tracking telescope mount proves

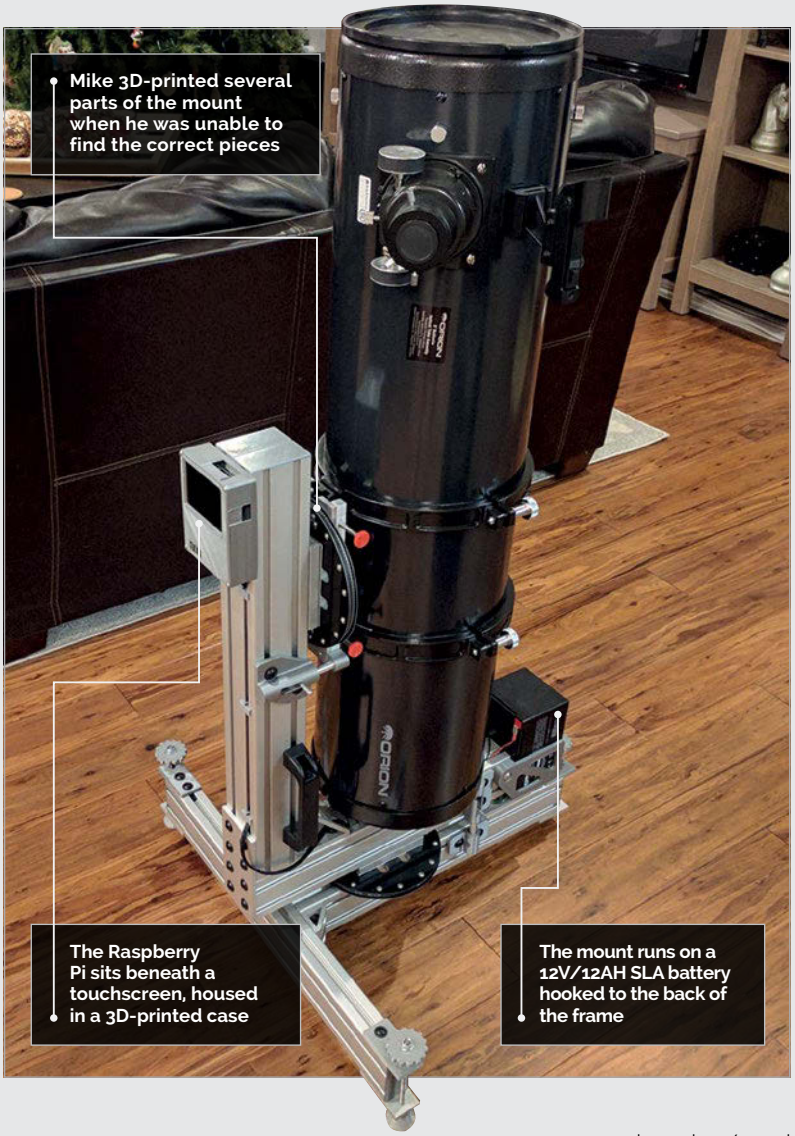
Quick Facts

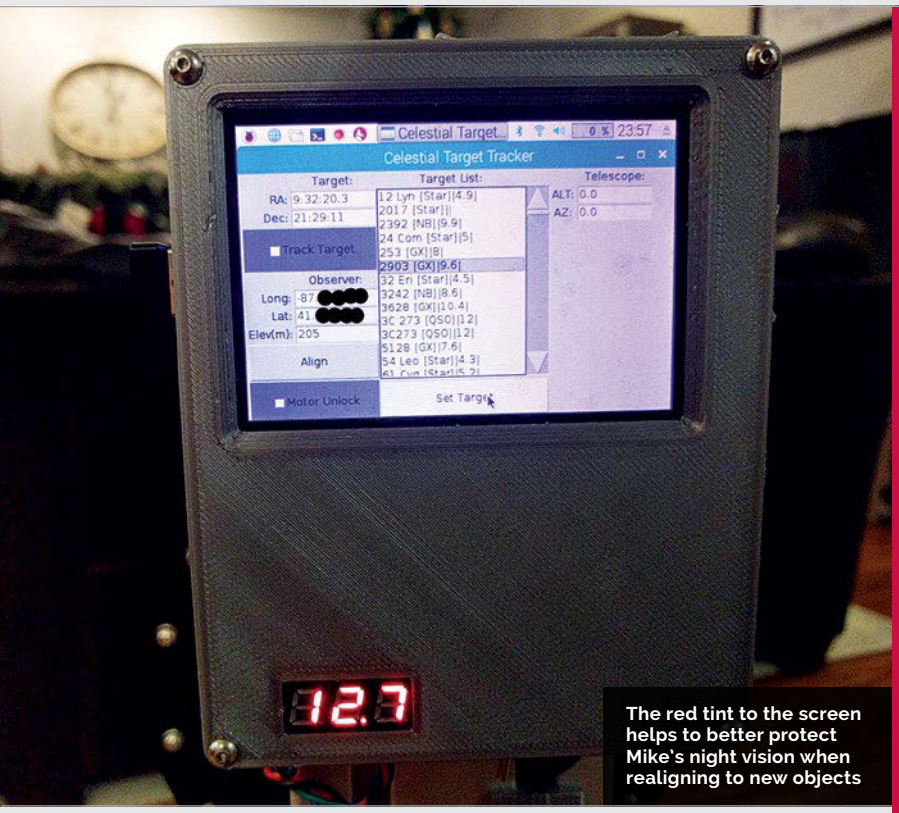
- Mike has no previous experience of coding Python
- The tracking mount is set to find 800 preset locations
- The mount runs on a battery, making it portable
- It cost $350-$400 to build
- See a video of the mount working here: **magpi.cc/2lGLgWf**

If you search for Mike Hamende's user name '8PumpkinDonuts' online, you'll soon realise that he's quite the craftsman. From a suspended lounger for his fiancée, to a wristwatch built completely from scratch, Mike is a true maker. So it's really no surprise that he'd eventually build a Raspberry Pi project that we could sink our teeth into. Despite his lack of previous Python knowledge, and ignoring the fact he could easily purchase what he'd decided to build, Mike produced a tracking telescope mount for his 8″ Orion reflector.

"This is definitely the most involved project I've ever completed," Mike tells us. "I'm aware there are mounts I could have bought to accomplish the same thing, but, while astronomy is a hobby of mine, building things is what I'm really into."

Mike used two Nema 17 stepper motors, powered by a Raspberry Pi 3, to angle the mount. For the frame, he used 3×3″ T-slot aluminium, utilising his CNC mill, CNC router, and CNC lathe to cut further pieces of aluminium for the mount.


Mike 3D-printed several parts of the mount when he was unable to find the correct pieces
The Raspberry Pi sits beneath a touchscreen, housed in a 3D-printed case
The mount runs on a 12V/12AH SLA battery hooked to the back of the frame



The red tint to the screen helps to better protect Mike's night vision when realigning to new objects

Two turntables with 60mm thrust ball bearings act as the basis for the motion mechanics, cut from 1/2″ thick aluminium. Mike then had to build his own drive pulleys using 3D-printed parts and fibreglass flanges. He amended technical documents he'd found online, removing teeth, until the pulley fitted perfectly within the build. From here, he also went on to print, rout, and fit all the pieces needed for the mount and frame before turning to a less-familiar subject: the Raspberry Pi.

> My hobby is making things and I've been doing it since I was old enough to hold a screwdriver

Mike originally created a script that would position the telescope to align with celestial objects by way of information from the US Naval Observatory. However, upon completing the code, he discovered PyEphem, a Python package that offered a much larger scope for celestial exploration. Though he had the PyEphem code, he couldn't find anything to control the stepper motors and linear acceleration, so he was still faced with the challenge of quickly learning enough Python to see him through.

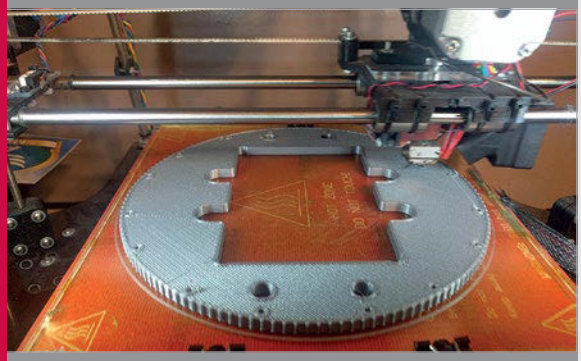
Once he'd cleared up his prototype breadboard and soldered the boards into place, Mike fitted the Pi with a touchscreen. The screen enables him to select one of 800 preset targets for the telescope to focus on. If he wants to view any other object, he simply uses a small wireless keyboard to enter the chosen coordinates manually.

Mike went to school to study mechanical engineering, though he attributes much of his skill to being a hobbyist. "My hobby is making things and I've been doing it since I was old enough to hold a screwdriver." He even converted his home lathe and router to be CNC, and built his own 3D printer too.

SCOPING THE SOLAR SYSTEM

>STEP-01
Creating the parts

Mike used his own tools to design and cut the appropriate parts for the mount out of 1/2″ aluminium. This allowed for greater control of the final look.



>STEP-02
3D-printing pulleys

Mike redesigned pre-existing technical documents to make them fit within the mount. With 288 teeth, this pulley was the largest he could print.

>STEP-03
Getting on the level

This large bullseye level allowed for less complication when aligning the telescope to the desired object, ensuring he was always on a flat surface.

DANIEL FERNANDEZ

A chemist, blogger, and consultant with a life-long passion for electronics and DIY projects.
github.com/danielfppps

Quick Facts

- It took two days for Daniel to develop the software
- Daniel wants to make an open-source voice assistant
- If you use the project, make sure the code launches at boot
- Daniel is an old hand with the Raspberry Pi, using it to automate several projects
- The case is to protect the Pi from sweat and spilt water

HEARTBEAT MONITOR

The Raspberry Pi is increasingly being used in health applications – here's how one maker made his own DIY Pi health device

The current surge of smart watches, however long it lasts, probably owes something to the activity trackers that are synonymous with anyone trying to keep track of their fitness. These devices track the user's heartbeat and use it to calculate things like periods of rest, calories burnt, and more. You can tell a lot of things from a heartbeat, so Daniel Fernandez has decided to create his own heartbeat monitor with help from a Raspberry Pi.

"I have always liked to monitor my heartbeat while I run on the treadmill," Daniel tells us. "There is simply no cheap, commercial solution that did everything I wanted: to plot the results in a graph that's easy to monitor, so that you can look at them while you run, and then save the data so that you can work with it later. Furthermore, I wanted some flexibility in how the data was plotted and how often the heartbeat was updated, and to be able to retrieve the data from the device in an easy-to-process format like CSV."

So began Daniel's build. He selected the Raspberry Pi 3 as it was a fully fledged computer with

You only need four items to make the monitor – how many projects are this simple?

built-in Bluetooth that could fit on his treadmill. He coupled it with a Polar H7 heart sensor and a 3.5″ screen, and 3D-printed a little case to fit it all in.

"It's not very difficult to build as most of the project is based on software," explains Daniel. "If you have the sensor, screen, and software, you could have it working in less than a few hours. I have made the software open source and available via a GitHub repository [**magpi.cc/2kBjoaM**]."

> "If you have the sensor, screen, and software, you could have it working in less than a few hours"

He's been successfully using it for its intended purpose, having run 20 miles with it since he finished the project.

Above The Pi waits patiently until it gets data from the heartbeat monitor

"So far, the Raspberry Pi has always worked correctly with my heartbeat sensor," Daniel says. "The program has always provided smooth plots and a very responsive display of my real-time heartbeat. I have been able to save my run data files and perform some more extensive mathematical analysis to better understand how my heart reacts to changes in intensity, cadence, inclination, and so on.

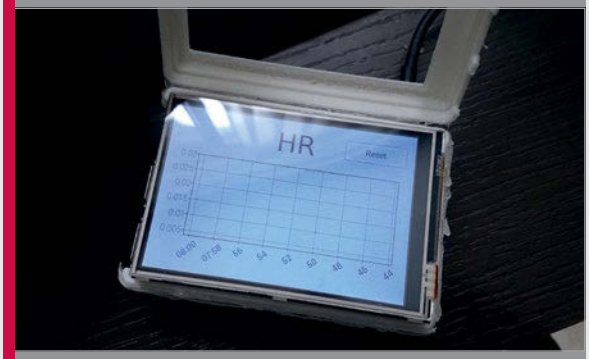
"I would want to include some functionality to save my heartbeat files to my Dropbox and even have the option to share the data on my social media accounts when I finish each run. I was also thinking about connecting the treadmill's control panel to the Raspberry Pi so that the treadmill could adjust intensity automatically to match a target heartbeat."

Other heartbeat sensors that follow the same standard as the Polar H7 should work with the project if you decide to make your own. Maybe it's a good excuse to keep up with your New Year's resolution?

PI AT HEART

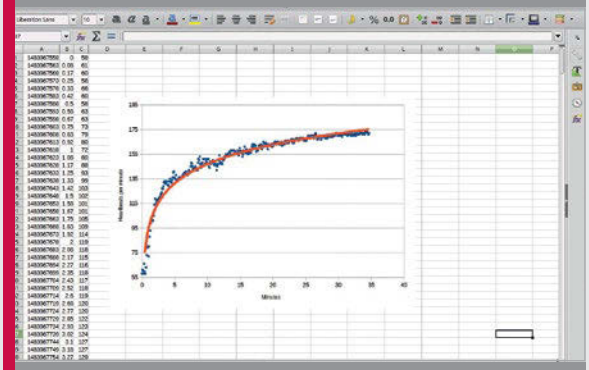
>STEP-01
Starting up

When the Pi boots up, it loads up a Python script that checks the heartbeat data from the sensor, and a Free Pascal program that plots the data.



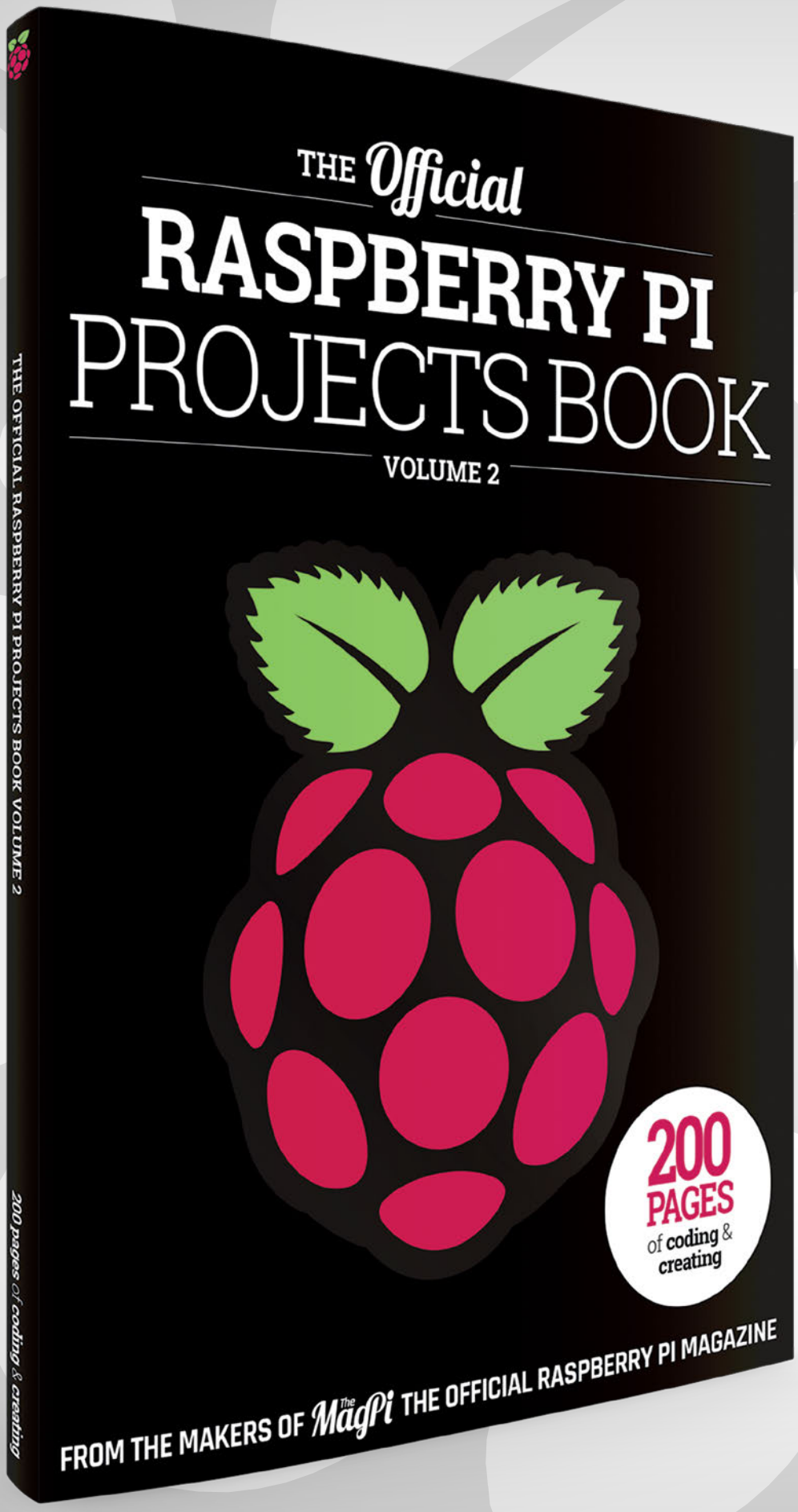
>STEP-02
Beat box

The heartbeat monitor automatically connects to the Pi and sends heartbeat data to the Python script. It's saved to a text file, which is used to plot the results.



>STEP-03
Saved

A reset button on the display resets the plot and saves the on-screen data to a CSV file for analysis at a later date.


THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 2
200
PAGES
of coding &
creating
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 2
200 pages of coding & creating

£12.99
200 pages of
Raspberry Pi

THE *Official*

RASPBERRY PI PROJECTS BOOK

VOLUME 2

Amazing hacking and making projects

from the creators of *The MagPi* magazine

Inside:

How to get started with Raspberry Pi

The most inspirational community projects

Essential tutorials, guides, and ideas

Expert reviews and buying advice

Available now

magpi.cc/MagPiStore

plus all good newsagents and:

WHSmith BARNES&NOBLE

Available on the App Store

GET IT ON Google Play

Available on Amazon

HOW TO BUILD AN HTPC MEDIA CENTRE

Build your home theatre media centre with a Raspberry Pi and OSMC

You'll Need

- OSMC
- Pi Zero or Raspberry Pi 2/3
- Keyboard for setup
- Smartphone for remote control

Building a media centre, or HTPC (home theatre PC) is one of the most rewarding projects a newcomer can complete.

It's easy to turn a Raspberry Pi into an HTPC, and you can quickly hook it up to your television.

Using an HTPC, you can play video and music files, and stream video and audio from online services. And unlike proprietary systems, such as Apple TV and Google Chromecast, you can use it to play just about any media format.

You can add streaming services to your HTPC, such as BBC iPlayer and Soma radio stations. These enable you to play media directly from the internet.

There are several pieces of software available for setting up an HTPC. Two options you'll find on the Raspberry Pi Foundation's Downloads page are OSMC and LibreELEC. Both are similar and run a media player called Kodi.

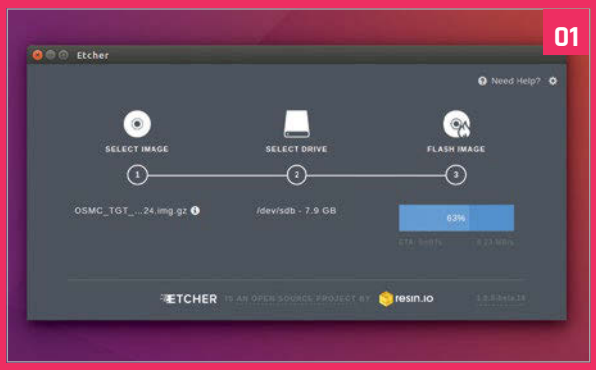
In this guide, we're going to opt for OSMC. It's a free and open-source media centre built on top of Debian, and is reliable and easy to use.

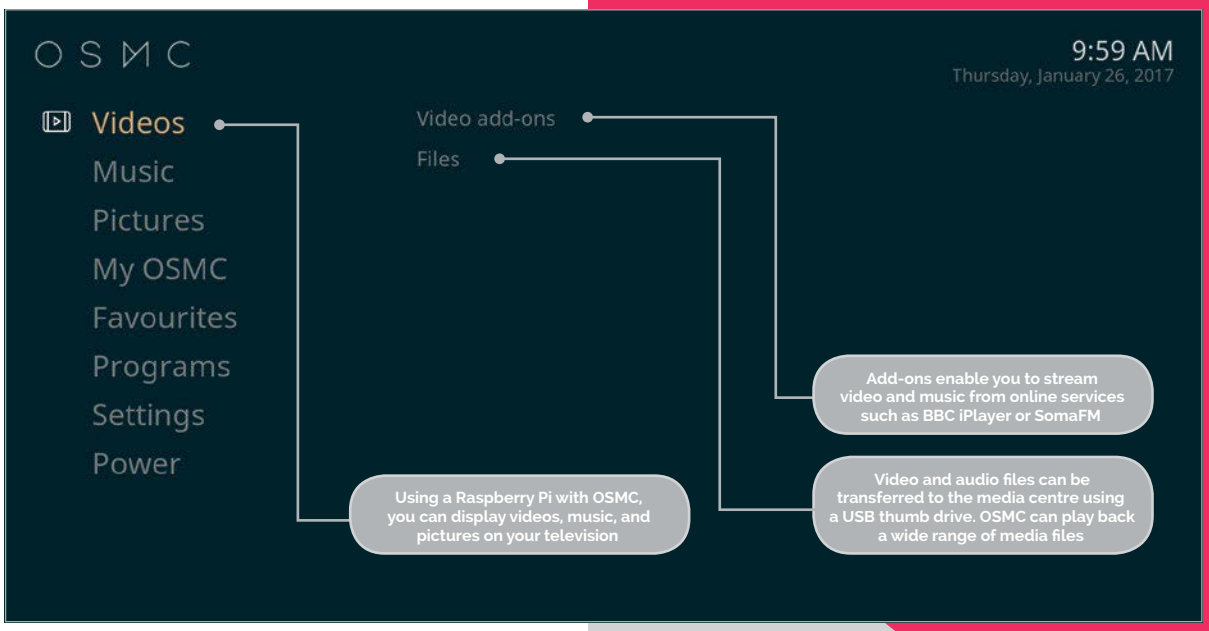
Setting up OSMC on a Raspberry Pi is a straightforward and rewarding project. Let's get started.

INSTALLING OSMC ON A RASPBERRY PI

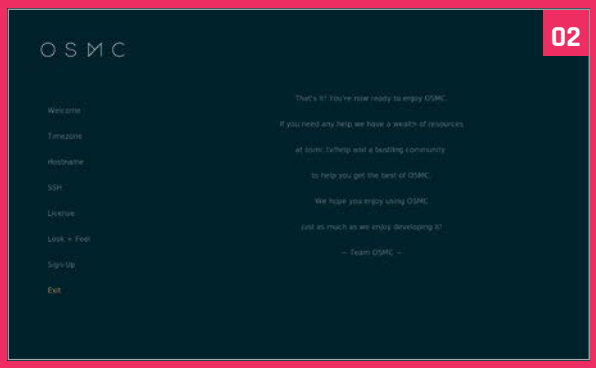
>STEP-01
Installation

First, install the OSMC image to an SD card. Visit **magpi.cc/2jqbw8R** in your web browser and download the OSMC disk image for Raspberry Pi Zero or Raspberry Pi 2/3. Flash the image to an SD card using Etcher (**etcher.io**).




OSMC
9:59 AM
Thursday, January 26, 2017
Videos
Music
Pictures
My OSMC
Favourites
Programs
Settings
Power
Video add-ons
Files
Add-ons enable you to stream video and music from online services such as BBC iPlayer or SomaFM
Video and audio files can be transferred to the media centre using a USB thumb drive. OSMC can play back a wide range of media files
Using a Raspberry Pi with OSMC, you can display videos, music, and pictures on your television

>STEP-02
Setup
Insert the SD card into your Raspberry Pi. Connect a keyboard and use an HDMI cable to connect a display. Power up and go through the Welcome section. It's a good idea to keep SSH enabled (the default option).

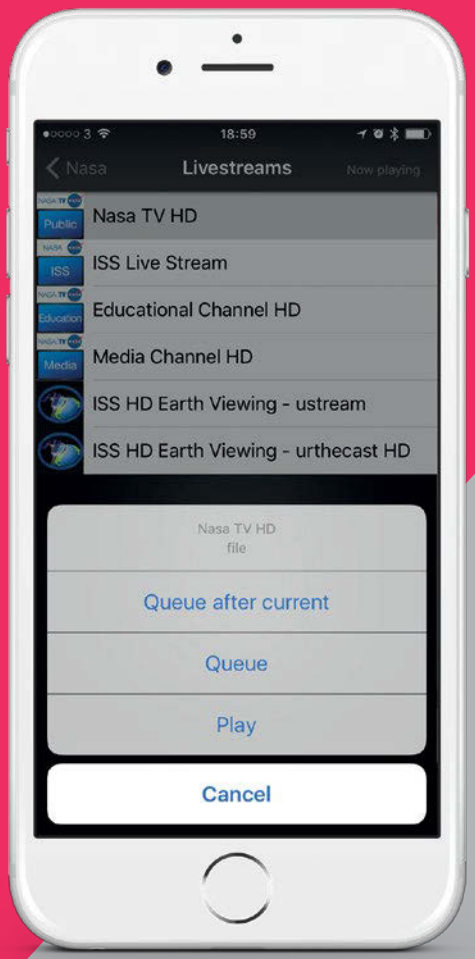


>STEP-03
Video files
To test video playback, we've downloaded a copyright-free video file of a movie called *Return of the Kung Fu Dragon* (**magpi.cc/2jxKryi**). Download the Ogg Video version and then transfer the file to a USB flash drive.

>STEP-04
Play video
Connect the flash drive to the Raspberry Pi. In the OSMC interface, select Videos > Files and the flash drive. Here you'll see the Ogg video file you copied across. Select it, and the video will start playing.

>STEP-05
Streaming services
Select Videos > Video Add-ons > Get More. Scroll down the services to find Nasa and select Install. Press **ESC** to get back to the main screen and select Videos > Nasa. Choose Livestreams and NASA TV to watch the station. For keyboard controls, visit the Kodi wiki (**magpi.cc/2jqjKoC**).

>STEP-06
Remote control
Open the app store on your phone and search for Official Kodi Remote (Kore on Android). Open the app, and click Add Host then Find Kodi. It should find the Raspberry Pi on your network. Click Save. Now use the app as a remote control.



HOW TO BACK UP YOUR RASPBERRY PI

Back up the files on a Raspberry Pi so you can quickly restore your files

You'll Need

- Raspberry Pi
- SD card
- Flash drive

One of the great things about using a Raspberry Pi is how easy it is to reset the whole system. If you're working on a project and you completely mess something up, it only takes minutes to reinstall the operating system and start again. Swift restoration is just one thing that makes the Raspberry Pi ideal for experimentation.

This constant state of being 'ready-to-refresh' can be a double-edged sword, though. The Raspberry Pi is a useful microcomputer: you can set up email and use the internet, write documents, install and use coding tools, and work on all manner of long-term projects.

If you use a Raspberry Pi for a long time, you'll want to think about a backup solution. A backup enables you to restore the Raspbian OS and your files.

Fortunately, there are several different ways to go about backing up the OS and your files.

Option 1: Copy the SD card image

The simplest way to back up your Raspberry Pi is to copy the entire SD card as an image.

This technique is the reverse of flashing your SD card when installing an OS to it. Instead of copying an image file from your computer to the SD card, you copy the entire SD card to an image file on your computer. This is, in fact, how image files are created in the first place.

Power down your Raspberry Pi and remove the SD card. Place it into an SD card reader and connect it to your computer.

Open a Terminal window on a Mac or Linux computer, and enter **df**. Take a look at what drives you have on your system. Now attach the SD card to your computer, and enter **df** again.

Spot the newly mounted drive: on a Linux machine, it will be something like /dev/sdb1, and on a Mac it will say /dev/disk2s1. The numbers may be different, so be sure to check carefully.

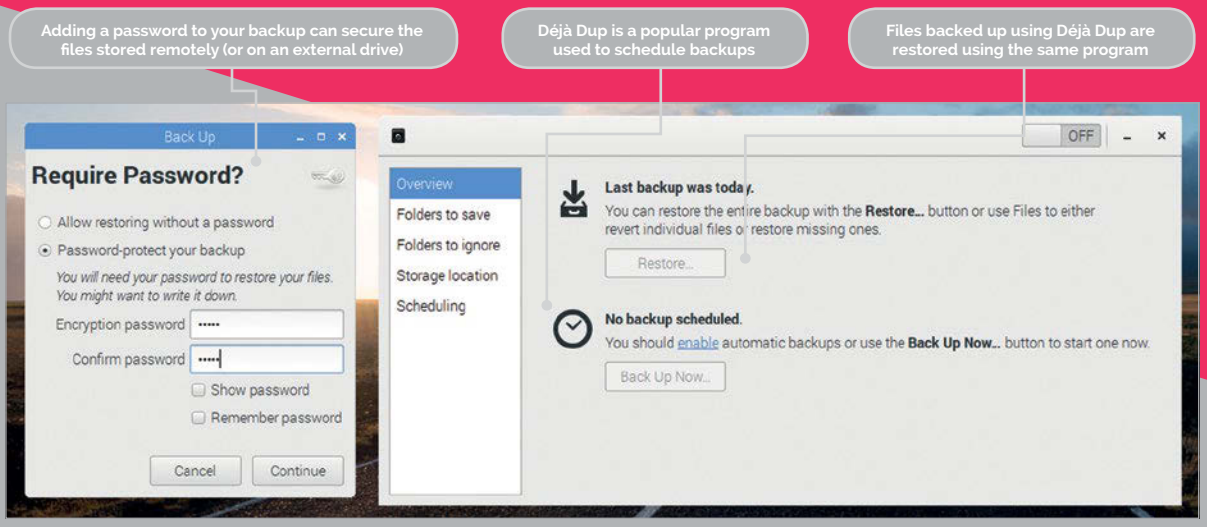
On Linux:

```
sudo dd bs=4M if=/dev/sdb of=raspbian.img
```

On a Mac:

```
sudo dd bs=4m if=/dev/rdisk2 of=raspbian.img
```

You can then use the **raspbian.img** file to restore your entire operating system (in its current state) to an SD card using **dd** in reverse, or by using an app such as Etcher (**etcher.io**) to flash the SD card.


Adding a password to your backup can secure the files stored remotely (or on an external drive)
Déjà Dup is a popular program used to schedule backups
Files backed up using Déjà Dup are restored using the same program
Back Up
Require Password?
Allow restoring without a password
Password-protect your backup
You will need your password to restore your files.
You might want to write it down.
Encryption password
Confirm password
Show password
Remember password
Cancel
Continue
OFF
Overview
Folders to save
Folders to ignore
Storage location
Scheduling
Last backup was today.
You can restore the entire backup with the Restore... button or use Files to either revert individual files or restore missing ones.
Restore...
No backup scheduled.
You should enable automatic backups or use the Back Up Now... button to start one now.
Back Up Now...

In Windows, you back up the SD card using Win32 Disk Imager (**magpi.cc/2bndEsr**).

Open the program and click Yes to the security alert window. Enter **C:\raspberrypi.img** in the Image File text box and click Read. The SD card will be written to the image file. When it says 'Read Successful', you can click OK.

Option 2: Back up the Home folder

The challenge with turning the SD card into an image file is doing it on a regular basis. You have to remove the SD card from your Raspberry Pi, attach it to a Mac or PC, and complete the whole backup.

The second backup option is to back up just your Home folder as a compressed file. Uncompressing the file enables you to browse and restore individual files and directories. Use the Terminal on your Raspberry Pi:

```
cd /home/
sudo tar czf pi_home.tar.gz pi
```

Copy the **pi_home.tar.gz** file to a USB flash drive for safe keeping.

Option 3: Schedule backups

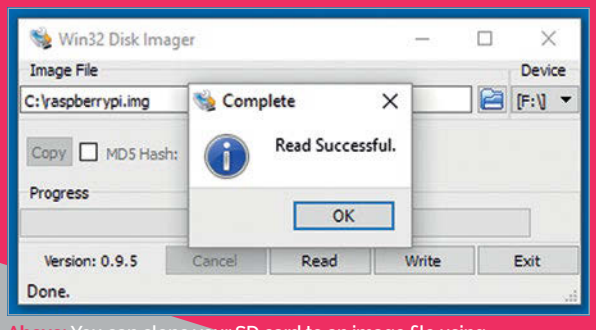
It is possible to schedule backups manually by scheduling the tar command as a cron job. See 'Scheduling Tasks with Cron' (**magpi.cc/2kg73Xs**).

Some Raspberry Pi users choose to use rsync (**magpi.cc/2kgchCH**) instead of tar, because this smartly copies updated files rather than the whole system. But you need to create an exclude file that ignores the contents of system folders. Take a look at this Stack Exchange discussion if you're interested (**magpi.cc/2kg1qIQ**).

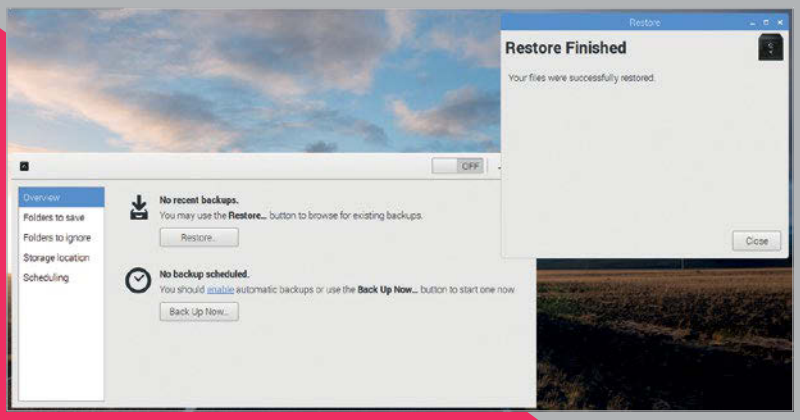
An easier approach is to use a free program called Déjà Dup. This automates rsync and gives it a user-friendly interface. It's an easy program to use, and you can back up your Raspberry Pi using Amazon S3, SSH, FTP, or by copying the files directly to a flash drive.

You can pick directories to include and exclude, and it's easy to restore backups. Déjà Dup is installed by default on Ubuntu, and is easy to install in Raspbian:

```
sudo apt-get update && sudo apt-get upgrade
sudo apt-get install deja-dup
```



Above: You can clone your SD card to an image file using dd on a Mac or Linux, and Win32 Disk Imager on a PC



Above: Files can be restored individually, or you can restore the whole of your Home folder to a fresh installation of Raspbian

Now you'll find a new program called Backups in Menu > Accessories.

Click on Folders to save and check that Home (pi) is listed. Folders to ignore should include Rubbish Bin.

Now click on Storage Location and change Amazon S3 to your flash drive.

Go back to Overview and click Back Up Now. Déjà Dup will start the first backup of your Home folder.

You'll be asked if you want to enter an encryption password. This is a good idea for security. Enter the same password in the Encryption Password and Confirm Password fields. Click Continue.

The first backup may take a while, depending on the size of your Home folder.

When Déjà Dup has finished backing up successfully, you can set up a schedule. Set the Off button to On. Click Scheduling and change Week to Day.

Backup will now store the contents of your Home folder to the flash drive on a daily basis, ready to be restored whenever you need it.

Restoring files

If you've had a total system failure and had to wipe your SD card, you'll need to reinstall Déjà Dup (follow the instructions earlier).

Make sure your backup flash drive is attached to the Raspberry Pi. Now open Accessories > Backups.

Click Restore to reinstall any missing files or directories. A 'Restore From Where?' window will open if you are installing from a fresh Raspbian image. Choose your flash drive using the Backup Location. Click Forward.

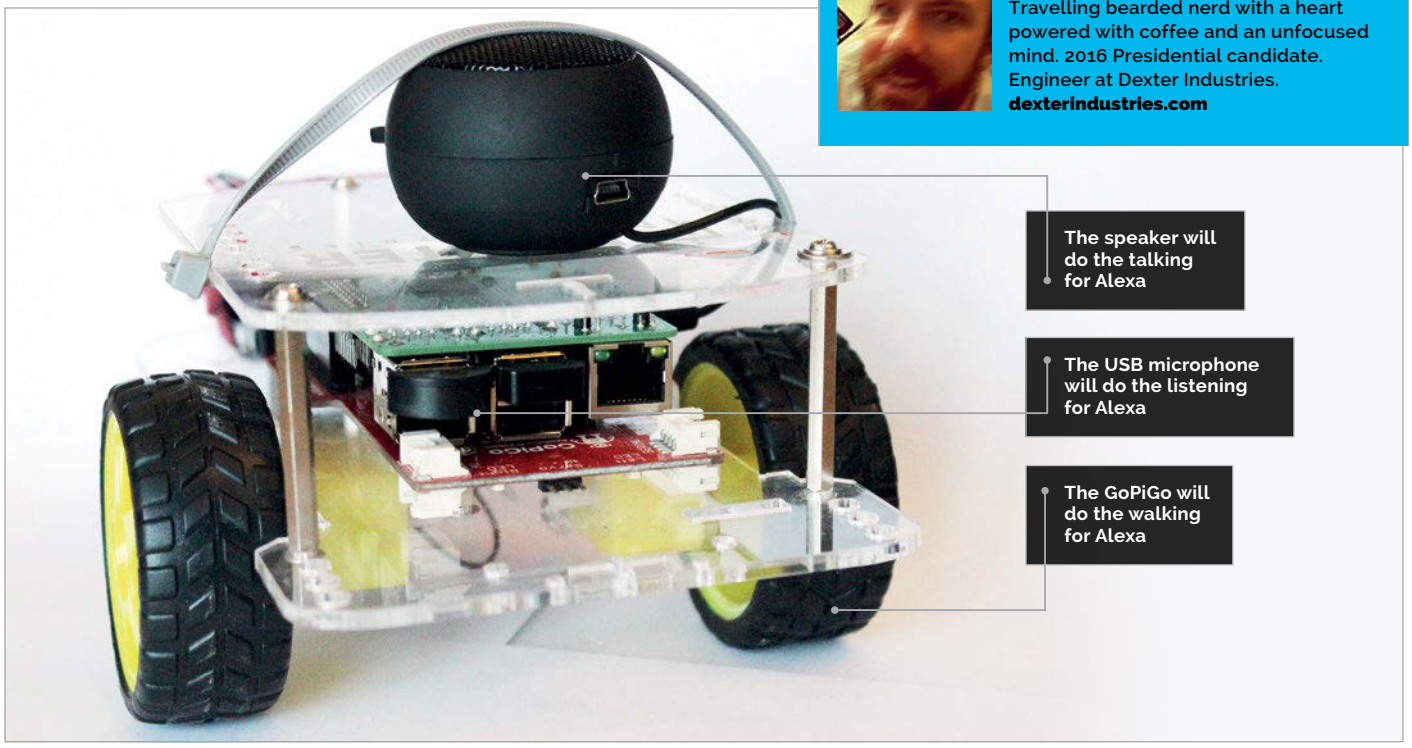
Choose the date to pick a backup from which you want to restore and click Forward. You have two options: restore files to original locations and restore to a particular folder. Choose to restore files to their original locations. Click Forward and Restore.

If you entered a password, you have to enter it in the Encryption Password field. Click Continue.

All the files from the Home folder in your previous backup will now be restored. You will see 'Restore Finished' when all your files are back. Click Close. You will now find all the files and directories from your Home folder reinstalled.

JOHN COLE

Travelling bearded nerd with a heart powered with coffee and an unfocused mind. 2016 Presidential candidate. Engineer at Dexter Industries.
dexterindustries.com

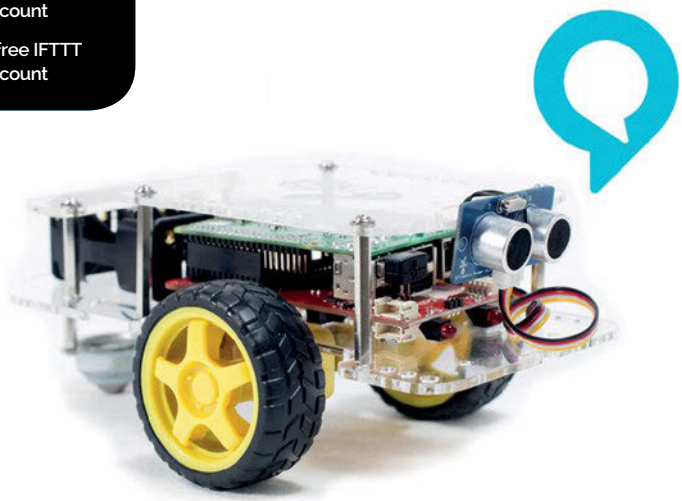

The speaker will do the talking for Alexa
The USB microphone will do the listening for Alexa
The GoPiGo will do the walking for Alexa

HOW TO MAKE AN ALEXA-POWERED ROBOT

Build an Amazon Alexa-controlled robot using the Raspberry Pi, allowing you to operate it with just your voice

You'll Need

- Raspberry Pi 3
- The GoPiGo **magpi.cc/2hA8f6i**
- Raspbian for Robots **magpi.cc/2kixPxB**
- Speaker for the Raspberry Pi **magpi.cc/2hAgCi7**
- Microphone **amzn.to/2g9mm1V**
- A free Amazon Developer account
- A free IFTTT account



Finally, a Raspberry Pi robot to do your bidding, controlled only by the sound of your voice! Amazon Alexa software can be put on the Raspberry Pi robot, the GoPiGo. In this project, we'll create an Alexa-based robot that will respond to voice commands, as well as answer your questions. You can ask Alexabot "How hot is it in Dubai?" or "What's the weather like in London?". What makes Alexabot really interesting is that you can order it around with your voice, using the Alexa Voice Service (AVS).

The first step towards robot world domination and building our Amazon Alexa-controlled robot is to set up AlexaPi.

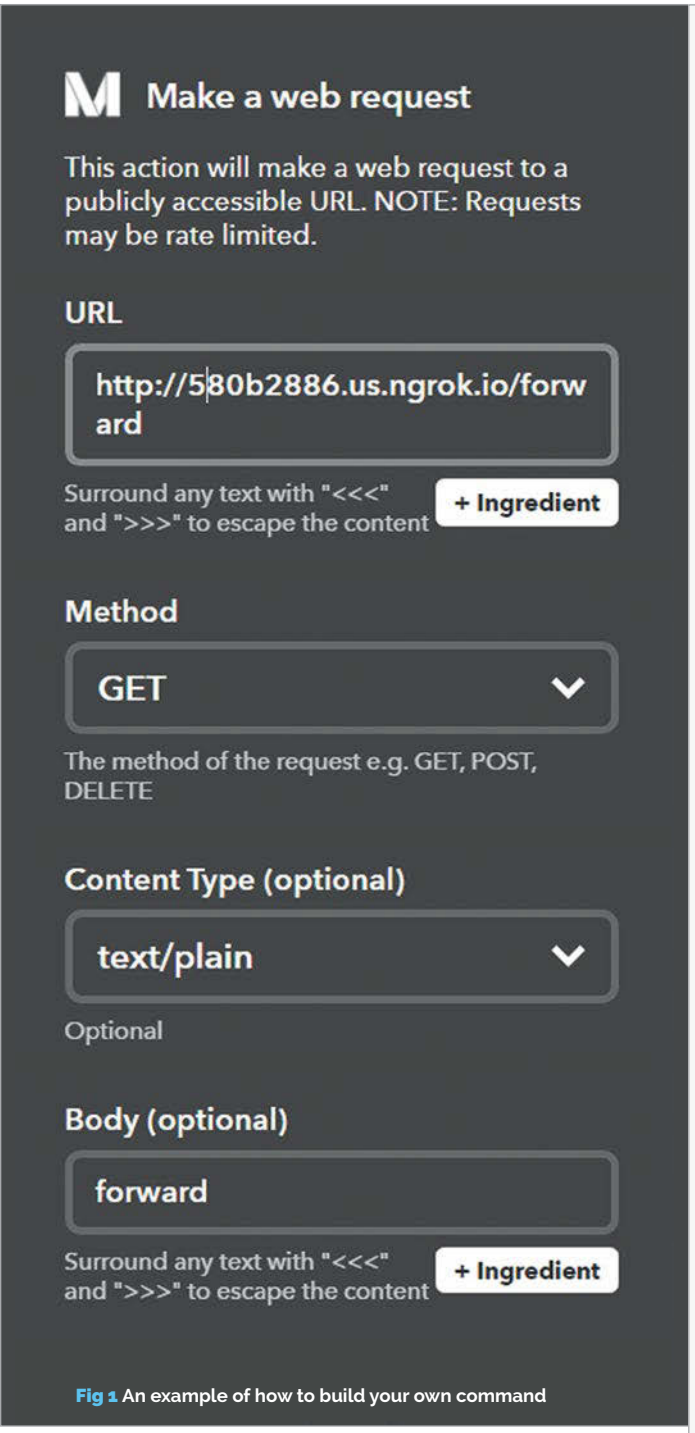

Make a web request
This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.
URL
http://580b2886.us.ngrok.io/forward
Surround any text with "<<<" and ">>>" to escape the content
+ Ingredient
Method
GET
The method of the request e.g. GET, POST, DELETE
Content Type (optional)
text/plain
Optional
Body (optional)
forward
Surround any text with "<<<" and ">>>" to escape the content
+ Ingredient

Fig 1 An example of how to build your own command

There are a few tutorials on how to turn your Raspberry Pi into an Amazon Echo using Amazon Alexa. However, AlexaPi (**magpi.cc/2kiyOxO**) is the easiest way to get Amazon Alexa on your Raspberry Pi. The project has prepared everything to set up Alexa Voice Services on your Pi in a streamlined and simple manner. The hardest part is setting up your Amazon Developer account and gathering your credentials (and really, that's not that hard!).

At this point, with your speaker and your microphone plugged into your Pi, you should be able to run Alexa on your GoPiGo, just as you would on an Amazon Echo. Ask it for a news update: "Alexa, what's the news?"

Alexabot.py

Language
>PYTHON

DOWNLOAD:
magpi.cc/2kiFlZz

```
from flask import Flask
import gopigo
import time

app = Flask(__name__)

@app.route('/')
def index():
  return 'Hello world'

@app.route('/forward')
def forward():
  print("Forward!")
  gopigo.fwd() # Send the GoPiGo Forward
  time.sleep(1)       # for 1 second
  gopigo.stop()       # then stop the GoPiGo
  return 'Alexabot moved forward!'

@app.route('/backward')
def backward():
  print("Backward!")
  gopigo.bwd() # Send the GoPiGo Backward
  time.sleep(1) # for 1 second
  gopigo.stop() # then stop the GoPiGo.
  return 'Backward!'

@app.route('/left')
def left():
  print("Left!")
  gopigo.left()
  time.sleep(1)
  gopigo.stop()
  return 'Left!'

@app.route('/right')
def right():
  print("Right!")
  gopigo.right()
  time.sleep(1)
  gopigo.stop()
  return 'Right!'

@app.route('/dance')
def dance():
  print("Dance!")
  for each in range(0,5):
    gopigo.right()
    time.sleep(0.25)
    gopigo.left()
    time.sleep(0.25)
    gopigo.bwd()
    time.sleep(0.25)
  gopigo.stop()
  return 'Dance!'

@app.route('/coffee')
def coffee():
  print("Coffee!")
  return 'coffee!'

if __name__ == '__main__':
  app.run(debug=True, host='0.0.0.0')
```

```
ngrok by @inconshreveable                                   (Ctrl+C to quit)

Session Status                connecting
Version                       2.1.18
Region                        United States (us)
Web Interface                 http://127.0.0.1:4041

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Fig 2 Your ngrok readout should look a bit like this

TRIGGER WORDS

Remember that your trigger words, the words that get Alexa going, start with "Alexa trigger".

Next, we'll set up IFTTT. If This Then That (IFTTT) is a service that lets us connect different parts of the web together. In our case, IFTTT helps us couple Alexa together with the Raspberry Pi. IFTTT lets us bridge the AlexaPi responses back to the Raspberry Pi. First, you'll need to set up an IFTTT account. (**ifttt.com/join**). Again, this is free.

Next, you'll need to add the IFTTT service to your AlexaPi. You can do this by first installing the Alexa app on your phone or tablet (**magpi.cc/2kyIJho**); it's available for iOS, Android, and Fire OS. With your phone on the same network as your Pi, connect to your AlexaPi and add the IFTTT service.

Next, in IFTTT, we will need an applet for each command we want to send to the GoPiGo. First, connect Amazon Alexa to your IFTTT account (**magpi.cc/2kizD9X**). Then we'll do the following to create an applet that will handle a single command:

01. Create an IFTTT applet (**magpi.cc/2jTYYn2**) for our first command. We will create an Alexa applet.
02. Select 'Say a Specific Phrase'.
03. Specify the command. Say "Backward".
04. Next, select 'That'.
05. We'll use the action service, Maker.
06. Click 'Make a web request' and specify the information.
07. The URL is either your provided ngrok URL or your custom domain. We will find this in our next step, but for now we can enter **ngrok.io**.
08. Our applet will be doing a 'GET' method, the content will be text, and don't put anything in the body.
09. Click 'Create Action' and then 'Finish'.

RUNNING THE FLASK SERVER

The Flask server runs in the background and needs to be set up in a different command-line window from the ngrok server.

Fig 1 shows how this is done in more detail. We will need to do this for all the moves you want your Amazon Alexa-controlled robot to make! In our project, we made seven: one each for Forward, Backward, Left, Stop, Right, Dance, and Coffee.

Next, we'll connect to the ngrok service. We will have IFTTT contact a server on the Pi, but the tricky part will be setting up a server on the Pi that's visible to the world outside our local wireless network. For this, you can use ngrok (**ngrok.com**), a service that allows you to connect to your Raspberry Pi through any network. IFTTT needs an internet-based URL to contact, so we'll use ngrok to make our server accessible to the outside world.

> "A robot that will respond to voice commands"

First, set up an account with ngrok. You can get away with a free account, but it will be much easier to set up Alexabot with a paid account. A paid account will allow you to set up named servers, rather than ngrok's randomly assigned server.

Next, install ngrok on your Raspberry Pi using the directions here: **ngrok.com/download**.

This should install all the software. You will next need to get your token, which will authorise your Raspberry Pi to ngrok. If you log into your account you will see step 2, with your token already populated. It should start with `./ngrok authtoken`. Copy the command, and paste it into your command line.

Test out ngrok

You can type the following into the command line:

```
./ngrok http 80
```

You should then see a server start up, as shown in **Fig 2**. When an IP address appears, try typing that IP address into your browser. You should see your default

server on Port 80 come up. We will start our server with the following command:

```
./ngrok http -subdomain=dexterindustries 5000
```

...where 'dexterindustries' is a reserved domain set up with our basic account on ngrok. If you go with the free account, you'll now need to go back and re-enter the domain name that ngrok gave your Pi into each of your IFTTT commands.

Set up the Flask server

The final step in getting our Amazon Alexa-controlled robot working is setting up the Flask server on the Raspberry Pi.

We'll set up a Flask server in Python to listen to IFTTT. To install Flask on the Raspberry Pi, enter the following into your command line:

```
sudo pip install flask
```

You can try running the Flask server we've provided in our GitHub code and see what happens:

```
python alexabot-flask-app.py
```

You should see the Flask server start!

To get an idea of how the code works on the Flask server, you can type the following in your web browser: **http://alexabot.ngrok.io/forward**. You should get this response in your browser: 'Alexabot moved forward!'.

This illustrates how the Forward command works; each command will work this way, with its own directory and custom code for moving the GoPiGo.

Now, with all the services set up, we should be able to say a command like "Alexa trigger forward"; Alexa will alert IFTTT, which will send an HTTP message through ngrok back to our GoPiGo and post to the web server running in Flask. The Flask program will command the GoPiGo to move forward.

The quick-start method to get running with Alexabot is to first start AlexaPi:

```
sudo python /opt/AlexaPi/src/main.py
```

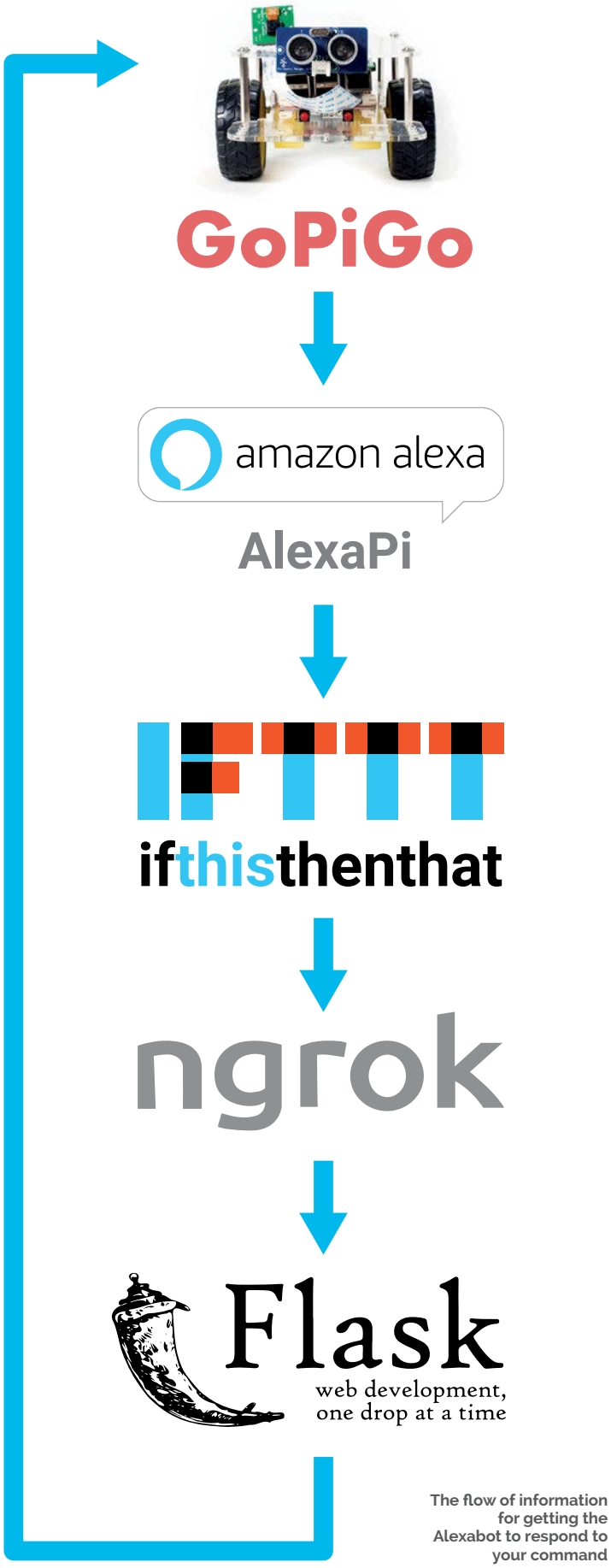
Next, start ngrok in a separate window:

```
sudo ~/ngrok/ngrok http
-subdomain=dexterindustries -log=stdout
5000 > log.txt &
```

Finally, start the Flask server:

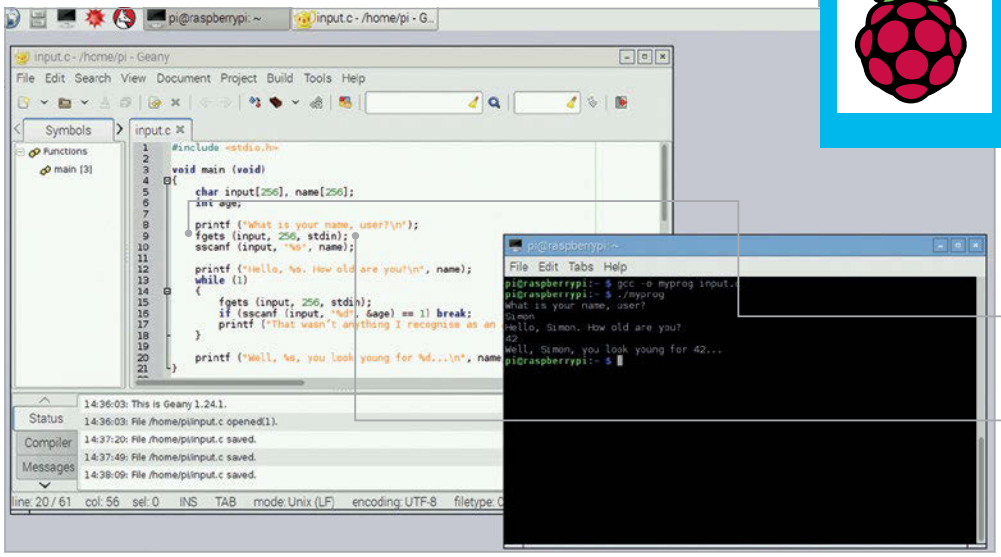
```
sudo python alexabot-flask-app.py
```

Start talking! Remember, your commands need to start with "Alexa trigger..." and then the command you want the robot to carry out.



The flow of information for getting the Alexabot to respond to your command

SIMON LONG
Works for Raspberry Pi as a software engineer, specialising in user interface design. In his spare time he writes apps for the iPhone and solves crosswords.
raspberrypi.org



AN INTRODUCTION TO C PART 09

USER INPUT

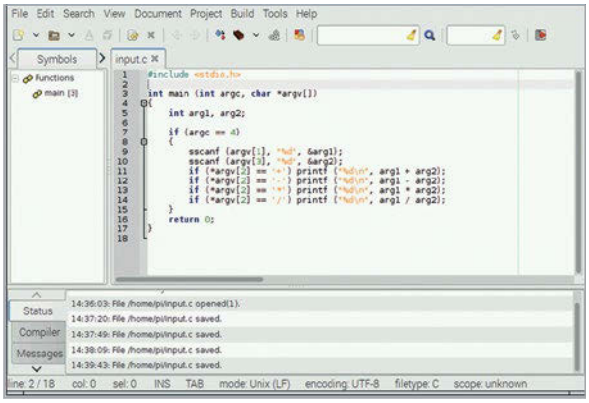
Reading and interpreting input from the user

In the previous instalments, we've looked at how to print program output to the terminal, but in order to interact with the user, this needs to be a two-way process. This instalment looks at how we can read and interpret input that the user enters in the terminal.

The **printf** is the standard way of writing formatted text output from a program, but how can we read what the user types in?

In the last instalment, we looked at the **sscanf** function, which reads values from a string. There's an equivalent function called **scanf**, which reads values typed in by the user:

```
#include <stdio.h>

void main (void)
{
  char input[256];
  int age;

  printf ("What is your name, user?\n");
  scanf ("%s", input);

  printf ("How old are you, %s?\n", input);
  scanf ("%d", &age);

  printf ("%s, you're %d?\n", input, age);
}
```



Above The argc and argv arguments to the main function can be used to access parameters typed on the command line when the program is run

scanf works exactly like **sscanf**, but has one fewer argument, as it reads from the keyboard rather than from a string.

However, it's not the best way of getting input, as it relies on the user doing what you expect, and this doesn't always happen! (For example, in the code above, if the user types in 257 characters when asked for their name, they will overflow the space allocated for the input string, and bad things may happen…)

A better way

A better approach is to read each line the user enters into a buffer string, and then use **sscanf** to read

SCANF

Just like sscanf, scanf returns an integer indicating how many values it successfully read, which you can use to check for errors. One problem is that scanf only removes matched values from the input buffer, so if a scanf fails to match anything, what the user typed will be read again on the next call to scanf. It really is easier to use fgets and sscanf!

GET THE NUMBER RIGHT

Remember that the first item in the argv array – argv[0] – is the name of the program itself, not the first parameter. The actual parameters start at argv[1].

values from that string. The C library function **fgets** is useful for this. See the following example:

```
#include <stdio.h>

void main (void)
{
  char input[256], name[256];
  int age;

  printf ("What is your name, user?\n");
  fgets (input, 256, stdin);
  sscanf (input, "%s", name);

  printf ("How old are you, %s?\n", name);
  while (1)
  {
    fgets (input, 256, stdin);
    if (sscanf (input, "%d", &age) == 1)
      break;
    printf ("Not an age, try again!\n");
  }

  printf ("%s, you're %d?\n", input, age);
}
```

fgets takes three arguments. The first is the buffer into which it should store the input. The second is the maximum number of bytes it will write into that buffer, which is useful to prevent overflow situations like that mentioned above. The third tells it where to read from – in this case, **stdin** (short for 'standard input') is the keyboard.

When we ask the user for input, we use **fgets** to read up to 256 characters of whatever they type (up to the point at which they press the **ENTER** key), and we then use **sscanf** to interpret it. When asking for the user's age, we use the value returned by **sscanf** to check that the user has entered what you expect, and loop until they give a valid answer. This makes it easier to handle the cases where the user types something unexpected!

Reading parameters

There's another way to get input to your program, which is to supply it as a parameter when you start the program from the command line. We've always shown the **main** function as

```
void main (void)
```

This works, as you've seen, but it isn't strictly correct. The strict definition of **main** looks like this:

```
int main (int argc, char *argv[])
```

But if you'd seen that in instalment 1, you'd have run a mile, wouldn't you?

main returns an integer – this is a success or failure code which some operating systems can use for processing in a shell script or the like. Traditionally, if a program succeeds, **main** returns 0; if it fails, it returns a non-zero error code. For programs that run on their own, you don't need to worry about it!

argc is the number of parameters given on the command line after the program name. This number includes the program name itself, so it's always 1 or more; if parameters are provided, it will be 2 or more.

char *argv[] is an array of strings, and each string is one of the parameters.

It's probably easier to understand that in practice, so here's an example:

```
#include <stdio.h>

int main (int argc, char *argv[])
{
  int param = 0;
  while (param < argc)
  {
    printf ("%d is %s\n",param,argv[param]);
    param++;
  }
  return 0;
}
```

Try running this as before, just by typing its name. Then try typing other things after the name on the command line and see what it prints.

Here's an example of a simple calculator written using parameters:

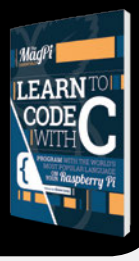
```
#include <stdio.h>

int main (int argc, char *argv[])
{
  int arg1, arg2;
  if (argc == 4)
  {
    sscanf (argv[1], "%d", &arg1);
    sscanf (argv[3], "%d", &arg2);
    if (*argv[2] == '+') printf ("%d\n", arg1 + arg2);
    if (*argv[2] == '-') printf ("%d\n", arg1 - arg2);
    if (*argv[2] == 'x') printf ("%d\n", arg1 * arg2);
    if (*argv[2] == '/') printf ("%d\n", arg1 / arg2);
  }
  return 0;
}
```

argv[2] is the second parameter; this is a string, but we're only interested in the first character, so we use ***argv[2]** to get this.

Make sure you separate the arguments from the operator with spaces so they are identified as separate parameters – **<progname> 2 + 2** rather than **<progname> 2+2**.

LEARN TO CODE WITH C BOOK

Want to learn more C? You can find the entire series and more in the *Learn to Code with C Essentials* book, available digitally and in print now at **magpi.cc/learn-c-book**



STDIN AND STDOUT

We talk about 'stdin' in this article, which is the 'standard input' stream – what the user types at the console. You may sometimes see references to 'stdout', which is the 'standard output' stream – as you might expect, this is output which is printed to the console, usually via printf.

GERMANO CESARI
CEO and Glise lead developer at twistednormal ltd. Writes code for a living, smokes ribs for pleasure.
twistednormal.com

MODEL A SQUID WITH GLISE

You'll Need

- Raspbian
- Glise **magpi.cc/2kqaliq**
- Experimental GL drivers **magpi.cc/2knRTvS**
- Three-button mouse

How to get started with 3D modelling and turn your Raspberry Pi into a personal graphic workstation

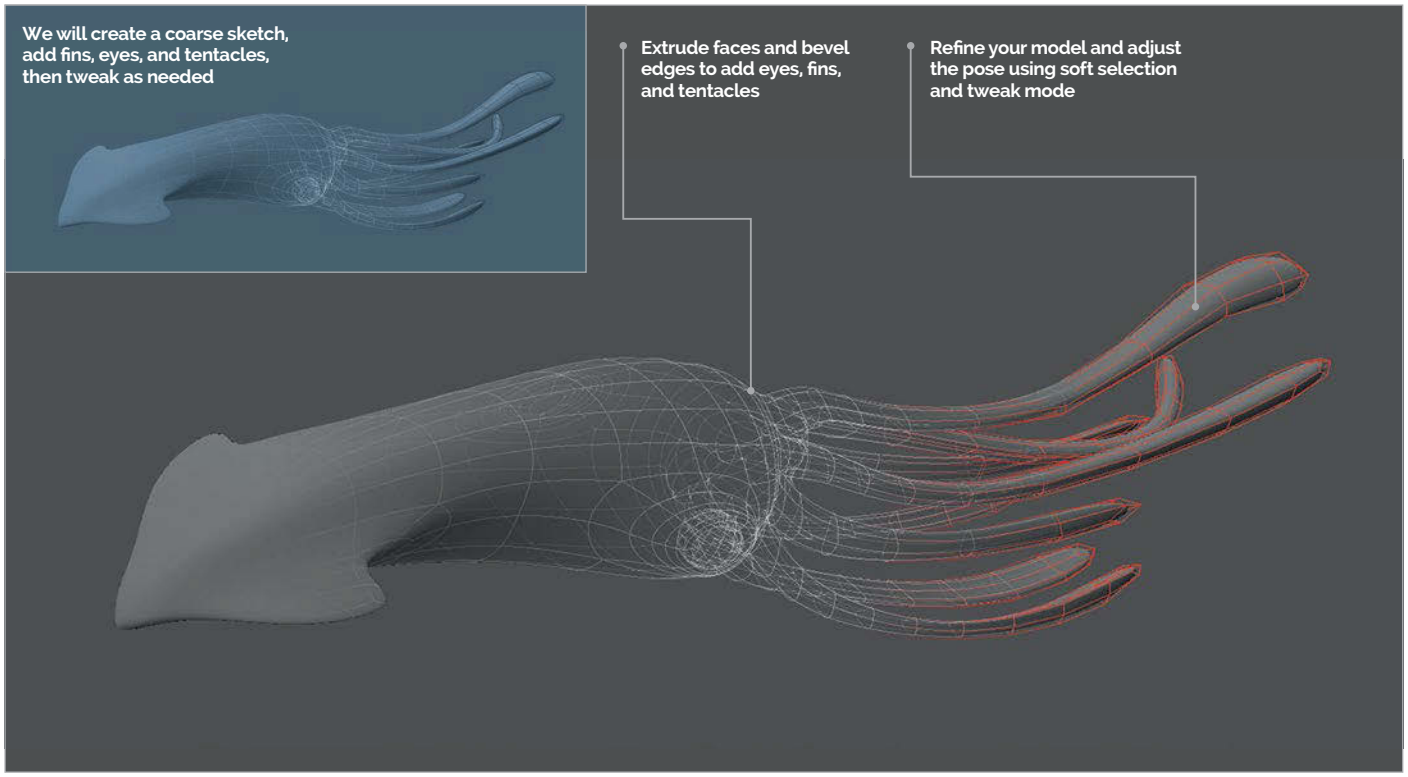
During the last few years, 3D applications have been behind many achievements in computational and video hardware, addressing whole new branches of computer graphics. Once known as 3D modelling applications, these packages are now best described as fully fledged modelling suites, with ever-higher hardware requirements and steeper learning curves. We want to take a step back and focus on pure 3D modelling using Glise, a new free modeller with minimal hardware requirements and a painless interface. Built from scratch, it turns a Raspberry Pi into a true modelling tool. So let's get started and see what your Raspberry Pi can do!

Set up and install

You must first activate the experimental OpenGL drivers – find out how on the Raspberry Pi website here: **magpi.cc/2knRTvS**. You might also want to increase the GPU memory to 128MB.

Download Glise for Raspberry Pi from **twistednormal.com**, right-click on the downloaded installer, and select Package Install.

Avoid any LXDE/Glise key binding conflict (they both use Alt+mouse drag). You can either edit **~/.config/openbox/lxde-pi-rc.xml** and comment out the whole `<context name="Frame">` section, or simply press the Function (Windows) key when using the Alt key to move the camera in Glise.


We will create a coarse sketch, add fins, eyes, and tentacles, then tweak as needed
Extrude faces and bevel edges to add eyes, fins, and tentacles
Refine your model and adjust the pose using soft selection and tweak mode

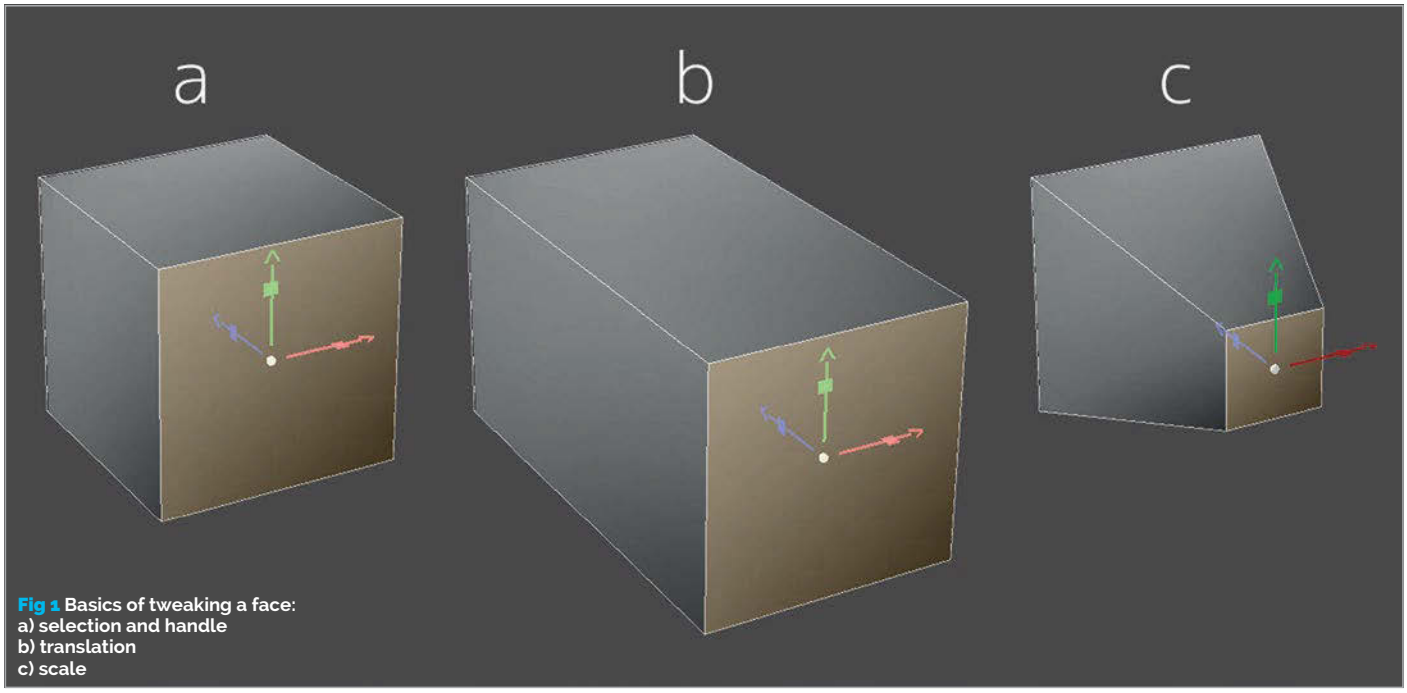


Fig 1 Basics of tweaking a face:
a) selection and handle
b) translation
c) scale

What we are going to do

3D modelling is an iterative process where you push, pull, and tweak vertices, edges, and faces. But how do you turn simple, geometrical primitives into an alien spider, a flying mussel, or a panda playing a banjo?

As any good artist, you will start by sketching out the basic proportions of the model using a very coarse mesh, which has only few components and allows you to make big changes quickly. As your modelling session progresses, you'll work on increasingly complex details, adding new, smaller components, either locally or by subdividing your whole model.

We're going to draw a squid, since they're relatively easy to model, yet complex enough to let us cover several fundamental concepts.

Modelling basics

Instead of going ahead with an endless list of actions to perform, like 'click there', 'open that' and the like, let's go through the fundamental concepts and procedures needed to sketch the squid head, which, surprisingly enough, you'll be able to model all by yourself once you've grasped the basics. You might want to check the online documentation at **magpi.cc/2knLzo2** (and you definitely should) while practicing, to get more in-depth information about the topics discussed here.

Glise's main window may contain several independent viewports, each holding either a scene manager, a Python console, or a modelling canvas (an ambient occlusion renderer is also available on Linux and Windows). Since we're not going to deal with scene management or Python programming, we will focus on the modelling canvas, which, as the name implies, is where all the magic happens. You can easily spot a modelling canvas from the little axis gizmo on its lower-right corner. By default, Glise will use a single, full-screen modelling canvas. If you want to change the default layout, check the online documentation at 'Layouts, Viewports and Dialogs'.

Creating a cube is straightforward: use a right mouse button (RMB) click anywhere in a modelling canvas to open the pop-up menu, and select Create > Cube. Now that there is an object in the scene, you can move the camera by pressing the **ALT** key and dragging the left mouse button (LMB) to rotate, the middle mouse button (MMB) to pan, or the RMB to zoom. Whenever the Alt key is pressed, drag the mouse to move the camera. Any time you want to centre the cube, just press Space and it will be framed in the current canvas. Practise with these controls: moving the camera and framing things must become second nature. See 'Navigation' in the online documentation for more camera controls.

> "Focus on the modelling canvas, which is where all the magic happens"

All this is fine and dandy, but you won't get very far just by rotating a camera around a cube. So we're going to tweak a face, which is, yes, modelling! Left-click on a face to select it. As the face turns yellow, a 3-axis handle is placed right on its centre (**Fig 1-a**). You can now LMB-drag any of the arrows to translate the selection along a direction (**Fig 1-b**), or LMB-drag the centre of the handle to move it freely. Try it! Dragging an arrow with the middle mouse button will make you scale the selection along a direction.

STICKY KEYS

Start a tool, keeping its shortcut pressed, and leave tool mode by releasing it. No need for a RMB click.

TWEAK MODE

Press Ctrl+Shift and LMB/MMB-drag to translate/scale a preselected component without selecting it first and using the handle.

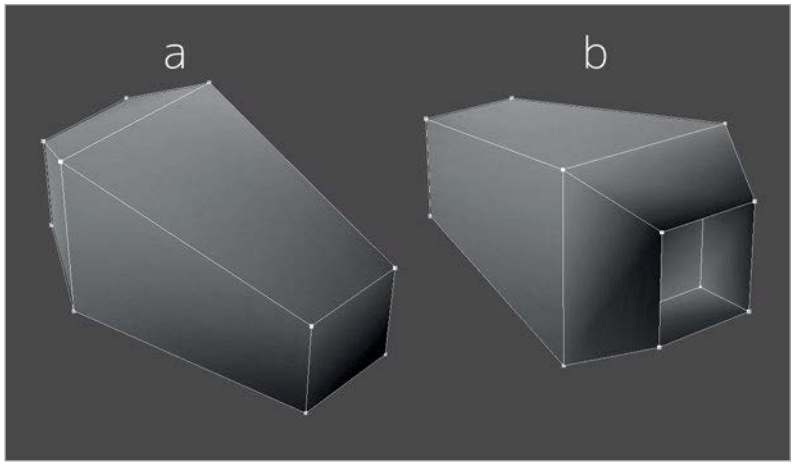


Fig 2 Coarse head sketch reference:
a) from the front
b) from the back

MMB-drag the centre of the handle for a uniform scale on every axis (**Fig 1-c**). Left-click on an empty area to clear the selection. Play with the handle and get comfortable with translating and scaling, change the type of components you can select and tweak (press A, S, D, G to switch to points, edges, faces, or mixed), and remember that a simple Ctrl+Z will undo any unwanted change. Check the online documentation at 'Preselection, Selection & Soft Selection' and 'Handles & Tweaking' to learn more about selections and tweaking.

> "When you model a base mesh, Glise automagically takes care of its subdivided surface"

Now one last, quick step: select a cube face and press X. You're currently using the extrude tool to perform a face extrusion. Move the mouse left and right to slide the newly created face and use a single right mouse button click to exit tool mode. Again, Ctrl+Z is your friend if you mess things up. Congratulations – you know a lot of stuff now!

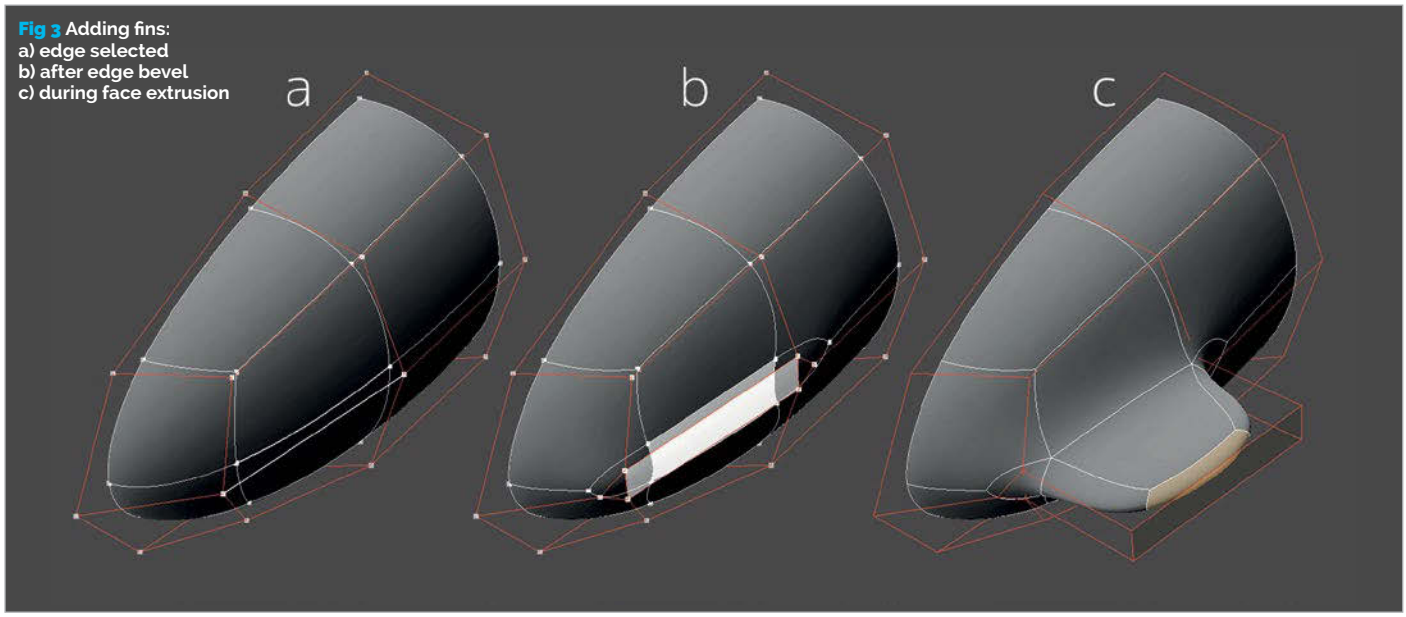
Sketching the head

So you know how to create a primitive, rotate the camera, tweak components, and extrude faces. Believe it or not, you can easily model our target shape in **Fig 2** all by yourself. You can skip this if you managed to model something close enough. Otherwise, this is how to proceed in five easy steps: create a cube, extrude a side face slightly outward, scale it smaller, and extrude it inward. Scale the opposite face smaller. Isn't modelling awesome?

We know what you're thinking: if that's a squid head, what on earth have we been eating? Don't worry; let us introduce you to the wonderful world of subdivision. Subdivision is the process of applying a specific set of rules to generate a more complex mesh from a simpler object. Glise handles each object as if composed of two distinct yet entangled entities: a base mesh and its subdivided surface. When you model a base mesh, Glise automagically takes care of its subdivided surface in real time. You may change the display mode of a modelling canvas by pressing 1 to see only base meshes, 2 to see only subdivided surfaces and work directly on them (the default), or 3 to see both.

Now press PgUp twice to increase the display approximation level of the surface of our sketch. Isn't it starting to look like a squid? Keep in mind that each level will eat four times more memory (and video memory) than the previous, so depending on the complexity of your model, you might not want to go further than a couple of levels. Use PgDn to get back to the previous level. See 'Introducing Glise' and 'Display' in the online documentation to learn more about what we've just covered.

Fig 3 Adding fins:
a) edge selected
b) after edge bevel
c) during face extrusion



Eyes, fins, and tentacles

Well, we're done now with our rough sketch, and it's time to proceed to the next level. We want to add a few details. Unfortunately, our base mesh (the one you actually model) is basically still very close to a simple cube. We can't extrude fins and tentacles from a cube's face, because we need components to work with, and in the middle of a face there's only, well, a face. So follow us, and everything will be clear in a minute: press F to switch to object selection mode, select our object, and press Shift+PgUp (aka 'Refine'). This will replace our base mesh with the first approximation level of its subdivided surface, which will keep its shape. Now our model has way more components, and we can have our fins in the right place. But wait: squids have a symmetrical body. Let's work on one side and have Glise take care of the other! We just have to properly set up symmetry on our squid: press S to switch to edge selection mode, and select one of the edges dividing the left side from the right. Press Alt+C (Symmetry > Compute From > Edges) and that's it. Isn't symmetry great? If instead you received an error message, check that the edge you selected really lies on the centre of the model. There are also other ways to compute symmetry – check 'Symmetry' in the online documentation to learn more.

Back to the fins: now that we've refined our model and enabled symmetry, select an edge like **Fig 3-a** and press B to use the edge bevel tool. Slide the mouse like in **Fig 3-b** and right-click to exit tool mode. Now select the newly created face and extrude it as in **Fig 3-c**. We'll make something similar for the tentacles, but we will work with vertices: press A to switch to vertex selection mode, select the vertices as in **Fig 4-a**, and press Ctrl+B to use the bevel tool on those. Right-click to exit tool mode when you're close to **Fig 4-b**. Then all we need to do is extrude the newly created faces as in **Fig 4-c**. Create the eye by using the bevel tool on a single vertex and deleting the resulting face with Del.

Now it's just a matter of tweaking a bit and working on a few further extrusions, and your final result will be very much like the squid in our picture. It doesn't have to be exactly the same – it must be what you want it to be.

What now?

Well, this is quite a complex subject, and this short tutorial wasn't meant to cover everything. There are tons of books, online tutorials, and videos on each single topic related to 3D modelling. Still, the tools you used in these pages can be found in any 3D software package, and the fact that you're doing this on your Raspberry Pi is awesome! Now draw and be creative, but above all, have fun!

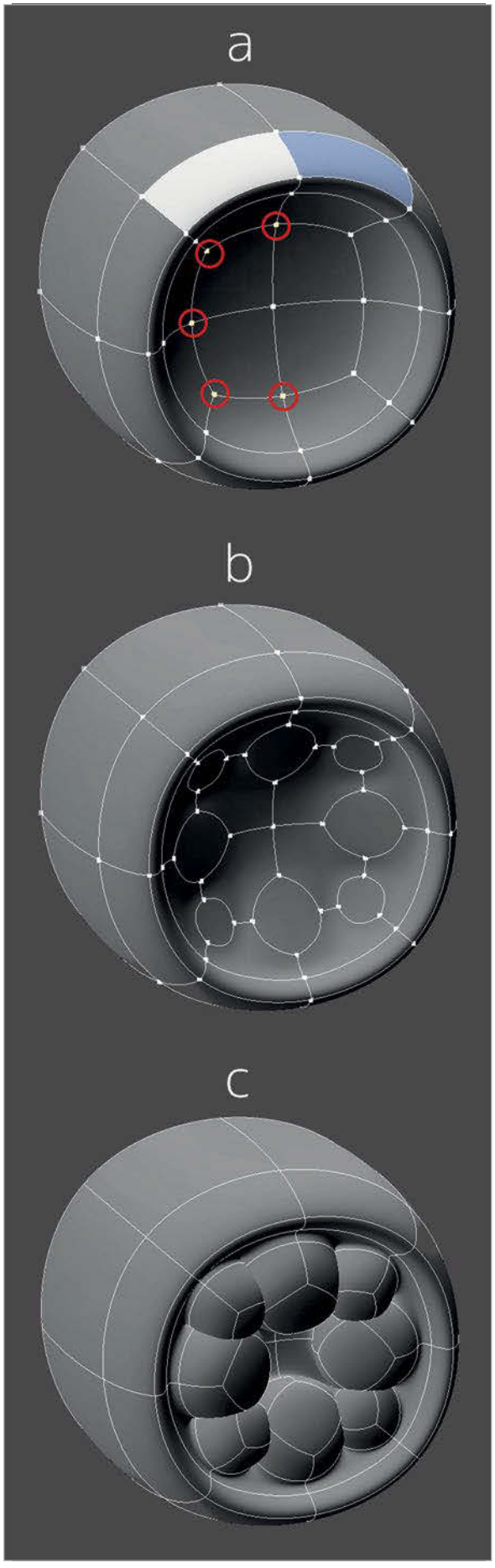


Fig 4 Adding tentacles:
a) vertices selected
b) after vertex bevel
c) during face extrusion

TOOLS AND PRESELECTION

Most tools also work on preselection. There is often no need for a selection to operate on single components.

LOCAL PIVOT

Local Pivot forces the camera to rotate around the centre of the scene, ignoring selections. Press L to toggle.

Sonic Pi PART 17

SAM AARON

Sam is the creator of Sonic Pi. By day he's a research associate at the University of Cambridge Computer Laboratory; by night he writes code for people to dance to. **sonic-pi.net**

SUBTRACTIVE SYNTHESIS

You'll Need

- Raspberry Pi running Raspbian
- Sonic Pi v2.9+
- Speakers or headphones with a 3.5mm jack
- Update Sonic Pi: `sudo apt-get update && sudo apt-get install sonic-pi`

Learn more ways to design sounds in Sonic Pi thanks to **Sam Aaron**

This is the second in a series of articles on how to use Sonic Pi for sound design. Last month we looked at additive synthesis, which we discovered was the simple act of playing multiple sounds at the same time to make a new combined sound. This month we'll look at a new technique commonly called subtractive synthesis, which is simply the act of taking an existing complex sound and removing parts of it to create something new. This is a technique which is commonly associated with the sound of analogue synthesizers of the Sixties and Seventies, but also with the recent renaissance of modular analogue synths through popular standards such as Eurorack.

Despite this sounding like a particularly complicated and advanced technique, Sonic Pi makes it surprisingly simple and easy – so let's dive right in.

Complex source signal

For a sound to work well with subtractive synthesis, it typically needs to be fairly rich and interesting. This doesn't mean we need something hugely complex – in fact, just a standard **:square** or **:saw** wave will do:

```
synth :saw, note: :e2, release: 4
```

Notice that this sound is already pretty interesting and contains many different frequencies above **:e2** (the second E on a piano), which add to create the timbre. If that didn't make much sense to you, try comparing it with the **:beep**:

```
synth :beep, note: :e2, release: 4
```

As the **:beep** synth is just a sine wave, you'll hear a much purer tone and only at **:e2** and none of the high crispy/buzzy sounds which you heard in the **:saw**. It's this buzziness and variation from a pure sine wave that we can play with when we use subtractive synthesis.

Filters

Once we have our raw source signal, the next step is to pass it through a filter of some kind, which will modify the sound by removing or reducing parts of it. One of the most common filters used for subtractive synthesis is something called a low-pass filter. This will allow all the low parts of the sound through, but will reduce or remove the higher parts. Sonic Pi has a powerful yet simple-to-use FX system that includes a low-pass filter, called **:lpf**. Let's play with it:

```
with_fx :lpf, cutoff: 100 do
  synth :saw, note: :e2, release: 4
end
```

If you listen carefully, you'll hear how some of that buzziness and crispiness has been removed. In fact, all the frequencies in the sound above note **100** have been reduced or removed, and only the ones below are still present in the sound. Try changing that **cutoff:** point to lower notes, say **70** and then **50**, and compare the sounds.

Of course, the **:lpf** isn't the only filter you can use to manipulate the source signal. Another important FX is the high-pass filter, referred to as **:hpf** in Sonic Pi. This does the opposite to **:lpf** in that it lets the high parts of the sound through and cuts off the low parts.

```
with_fx :hpf, cutoff: 90 do
  synth :saw, note: :e2, release: 4
end
```

Notice how this sounds much more buzzy and raspy now that all the low-frequency sounds have been removed. Play around with the cutoff value – note how lower values let more of the original bass parts of the source signal through and higher values sound increasingly tinny and quiet.

Low-pass filter

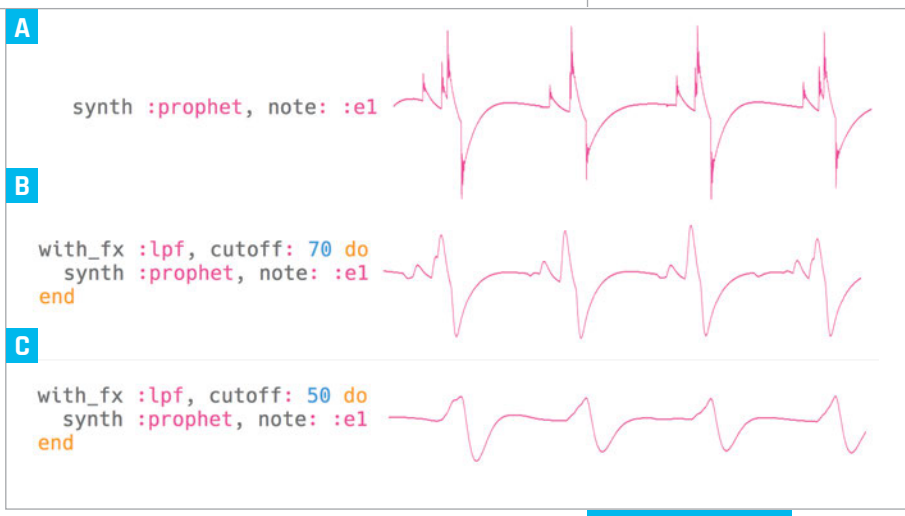
The low-pass filter is such an important part of every subtractive synthesis toolkit that it's worth taking a deeper look at how it works. This diagram (top right) shows the same sound wave (the **:prophet** synth) with varying amounts of filtering. At the top, section A shows the audio wave with no filtering. Notice how the wave form is very pointy and contains lots of sharp edges. It is these hard, sharp angles that produce the high crispy/buzzy parts of the sound. Section B shows the low-pass filter in action – notice how it is less pointy and more rounded than the wave form above. This means that the sound will have fewer high frequencies, giving it a more mellow, rounded feel. Section C shows the low-pass filter with a fairly low cutoff value – this means that even more of the high frequencies have been removed from the signal, resulting in an even softer, rounder wave form. Finally, notice how the size of the wave form, which represents the amplitude, decreases as we move from A to C. Subtractive synthesis works by removing parts of the signal, which means that the overall amplitude is reduced as the amount of filtering that is taking place increases.

Filter modulation

So far we've just produced fairly static sounds. In other words, the sound doesn't change in any way for the entirety of its duration. Often, you might want some movement in the sound to give the timbre some life. One way to achieve this is via filter modulation – changing the filter's options through time. Luckily, Sonic Pi gives you powerful tools to manipulate an FX's opts through time. For example, you can set a slide time to each modulatable opt to specify how long it should take for the current value to linearly slide to the target value:

```
with_fx :lpf, cutoff: 50 do |fx|
  control fx, cutoff_slide: 3, cutoff: 130
  synth :prophet, note: :e2, sustain: 3.5
end
```

Let's take a quick look at what's going on here. Firstly, we start a **:lpf** FX block as normal with an initial **cutoff:** of a low **50**. However, the first line also finishes with the strange **|fx|** at the end. This is an optional part of the **with_fx** syntax which allows



you to directly name and control the running FX synth. Line 2 does exactly this and controls the FX to set the **cutoff_slide:** opt to **3** and the new target **cutoff:** to be **130**. The FX will now start sliding the **cutoff:** opt's value from **50** to **130** over a period of three beats. Finally, we also trigger a source signal synth so we can hear the effect of the modulated low-pass filter.

Language
>RUBY

Above How the waveform alters when a low-pass filter is added with two cutoff values

Bringing it all together

This is just a very basic taster of what's possible when you use filters to modify and change a source sound. Try playing with Sonic Pi's many built-in FX to see what crazy sounds you can design. If your sound feels too static, remember you can start modulating the options to create some movement.

Let's finish by designing a function which will play a new sound created with subtractive synthesis. See if you can figure out what's going on here – and, for the advanced Sonic Pi readers out there, see if you can work out why we wrapped everything inside a call to **at** (please send answers to **@samaaron** on Twitter).

```
define :subt_synth do |note, sus|
  at do
    with_fx :lpf, cutoff: 40, amp: 2 do |fx|
      control fx, cutoff_slide: 6, cutoff: 100
      synth :prophet, note: note, sustain: sus
    end
    with_fx :hpf, cutoff_slide: 0.01 do |fx|
      synth :dsaw, note: note + 12, sustain: sus
      (sus * 8).times do
        control fx, cutoff: rrand(70, 110)
        sleep 0.125
      end
    end
  end
end

subt_synth :e1, 8
sleep 8
subt_synth :e1 - 4, 8
```

BEN NUTTALL
Raspberry Pi's community manager, creator of the GPIO Zero library, Jam master, and the Foundation's resident Python expert!
twitter.com/ben_nuttall

BUILD LOGIC GATES WITH GPIO ZERO SOURCE/VALUES

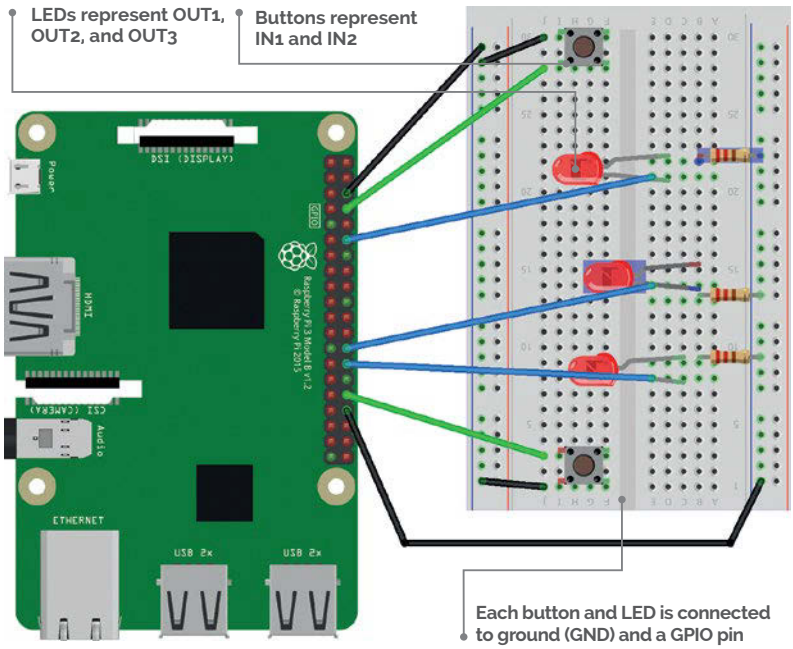
You'll Need

- The latest version of GPIO Zero **magpi.cc/2kmDu5l**
- Full-size breadboard
- 3 × LEDs
- 3 × 330Ω resistors
- 2 × push buttons
- 7 × male-to-female jumper wires
- 2 × male-to-male jumper wires

Learn how you can use source/values in GPIO Zero to easily connect devices together using a declarative style of programming

Source/values is a more advanced feature of GPIO Zero, but it's incredibly powerful and much simpler than you'd think! You can do more with less code, and it saves you from using `while` loops to keep updating the state of your devices. This clever concept allows you to describe how devices should behave, making your code more concise.

Source/values is easy to get started with, and takes a keen mind to master. There's some magic going on beneath the hood! You will learn how it all works, and put it into action by creating logic gates (AND, OR, and NAND), using LEDs and buttons.


LEDs represent OUT1, OUT2, and OUT3
Buttons represent IN1 and IN2
Each button and LED is connected to ground (GND) and a GPIO pin

What is source/values?

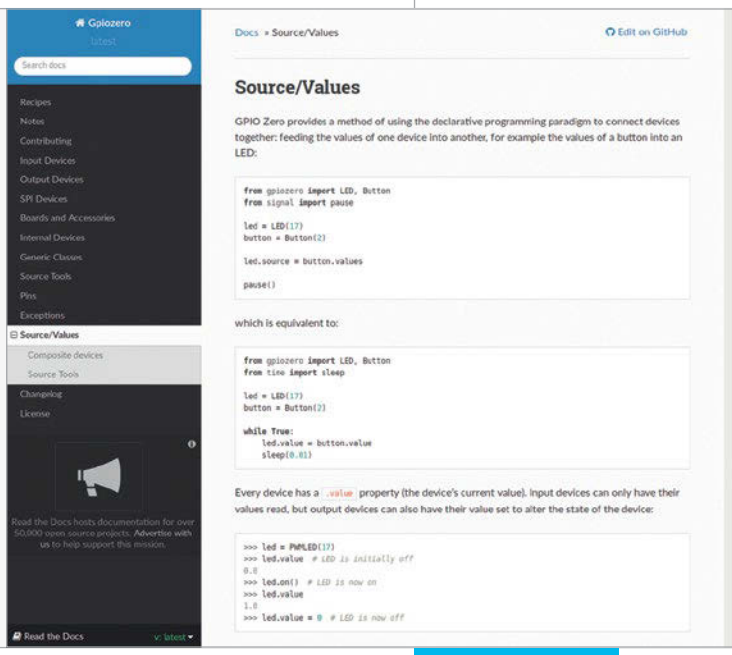
GPIO Zero is more than just a friendly API: it comes with some powerful tools for helping you program your next project. While you can start with simple procedural code, where each line is followed in sequence, you can progress to event-driven programming, meaning various events can be triggered in any order, or at the same time. And finally, the darkest of dark arts, source/values brings forth a third programming paradigm: declarative. This means you describe an object's behaviour in one line, and it will follow that instruction in the background. This is really handy if you have lots of things going on at once!

Source/values is the name for the concept in GPIO Zero for telling a device how it should behave, or for connecting up multiple devices together so they act in a particular way. Every GPIO Zero device has a `.value` property which you can read at any time: a button will tell you if it's pressed (True) or released (False), and an LED will tell you if it's lit (True) or not (False). Output devices (like LEDs) can have their value set; `led.value = True` will light the LED, just like using `led.on()`.

Every device also has a `.values` property, which is an iterator constantly yielding the current value. Output devices have a `.source` property which is where you tell the device where it should get its values from. Commonly, this will be another device's `.values` property, which is an easy way of connecting two devices together. For example, `led.source = button.values` means the LED will be on when the button is pressed.

source_values.py

```
from gpiozero import LED, Button
from gpiozero.tools import all_values, any_values, negated
from signal import pause

in_1 = Button(0)
in_2 = Button(1)

out_1 = LED(2)
out_2 = LED(3)
out_3 = LED(4)

out_1.source = all_values(in_1.values, in_2.values)
out_2.source = any_values(in_1.values, in_2.values)
out_3.source = negated(out_1.values)

pause()
```


Gpiozero
Docs » Source/Values
Edit on GitHub
Source/Values
GPIO Zero provides a method of using the declarative programming paradigm to connect devices together: feeding the values of one device into another, for example the values of a button into an LED:
which is equivalent to:
Every device has a .value property (the device's current value). Input devices can only have their values read, but output devices can also have their value set to alter the state of the device:
Read the Docs

The magic under the hood

This concept was developed by Dave Jones (author of the picamera library and co-developer of GPIO Zero) based on an idea formed in a GitHub issue (**rpf.io/gpz76**). The way it works is that when a device's source is set, a background thread is created to constantly set the device's value to the next item in the iterator. If source is set to another device's **.values**, this is effectively always reading the other device's current value. It doesn't have to be, though: it can be any iterator, or you can write your own function to send values to the device.

A handy thing about GPIO Zero is the fact that devices tend to have a standard value range – usually a sliding scale from 0 to 1 (or True and False, which are effectively just 0 and 1). This means that you can pass the values from one device into another: like a potentiometer (0–1) into a PWM LED (0–1) to dial up the brightness. Exceptions include **Motor**, which goes from –1 to 1 (–1 being full speed backwards, 1 full speed forwards), and composite devices like **Robot**, which is a tuple (–1, –1) to (1, 1) for left and right motor speeds.

Why is it useful?

Using this method of connecting devices is much more concise. Rather than using a **while** loop (five lines) to constantly check if a button is pressed to turn an LED on or off, or using an event-driven approach (two lines: **when_pressed** and **when_released**), one line tells it to match the button: **led.source = button.values**.

As well as connecting two devices directly together, you can process the value in between, using a custom function. GPIO Zero provides a set of commonly used tools for this kind of processing. Importing the negated function from **gpiozero.tools** allows you to specify that an LED should be lit when the button isn't pressed: **led.source = negated(button.values)**. See more source tools in the docs at **rpf.io/st**.

Programming logic gates

Let's begin. First, wire up two buttons and three LEDs on a breadboard. You'll need to import the gpiozero **LED** and **Button** classes, and the **all_values**, **any_values**, and **negated** source tools. Create objects representing each of the LEDs and buttons, providing the pin numbers they're wired to. The buttons represent two inputs; the LEDs three outputs. You're going to program the first LED to represent the binary AND of the two inputs, the second to represent the OR, and the third the NAND.

To define the AND of the two buttons, you'll need to use the **all_values** function from **gpiozero.tools**; **all_values** yields True if all of the inputs are True, which is an AND gate:

```
out_1.source = all_values(in_1.values, in_2.values)
```

To define the OR of the two buttons, you can use the **any_values** source tool; **any_values** yields True if any of the inputs are True, like an OR gate:

```
out_2.source = any_values(in_1.values, in_2.values)
```

You could define the NAND button using a combination of **negated** and **all_values**, but since you have already calculated the AND, you can simply read the state of the AND LED and negate that:

```
out_3.source = negated(out_1.values)
```

Finally, if you're running this code from a file rather than a Python shell, the last line, **pause()**, will keep your script running. Run your code and you'll see the NAND LED lit up. That's because (with no buttons pressed), it's the only gate that's active according to the rules. Try pressing the buttons in different combinations to check your logic! Now try programming some more logic gates: XOR, NOR, and XNOR.

Language
>PYTHON

DOWNLOAD:
magpi.cc/ZeroSourceValues

Above Read up on source/values in the gpiozero documentation at **rpf.io/sv**

CARRIE ANNE PHILBIN

Carrie Anne is the director of education at the Raspberry Pi Foundation and also the author of *Adventures in Raspberry Pi*.
raspberrypi.org

PIONEERS PROJECT

Enter the Pioneers challenge here:
magpi.cc/2iHKIP5

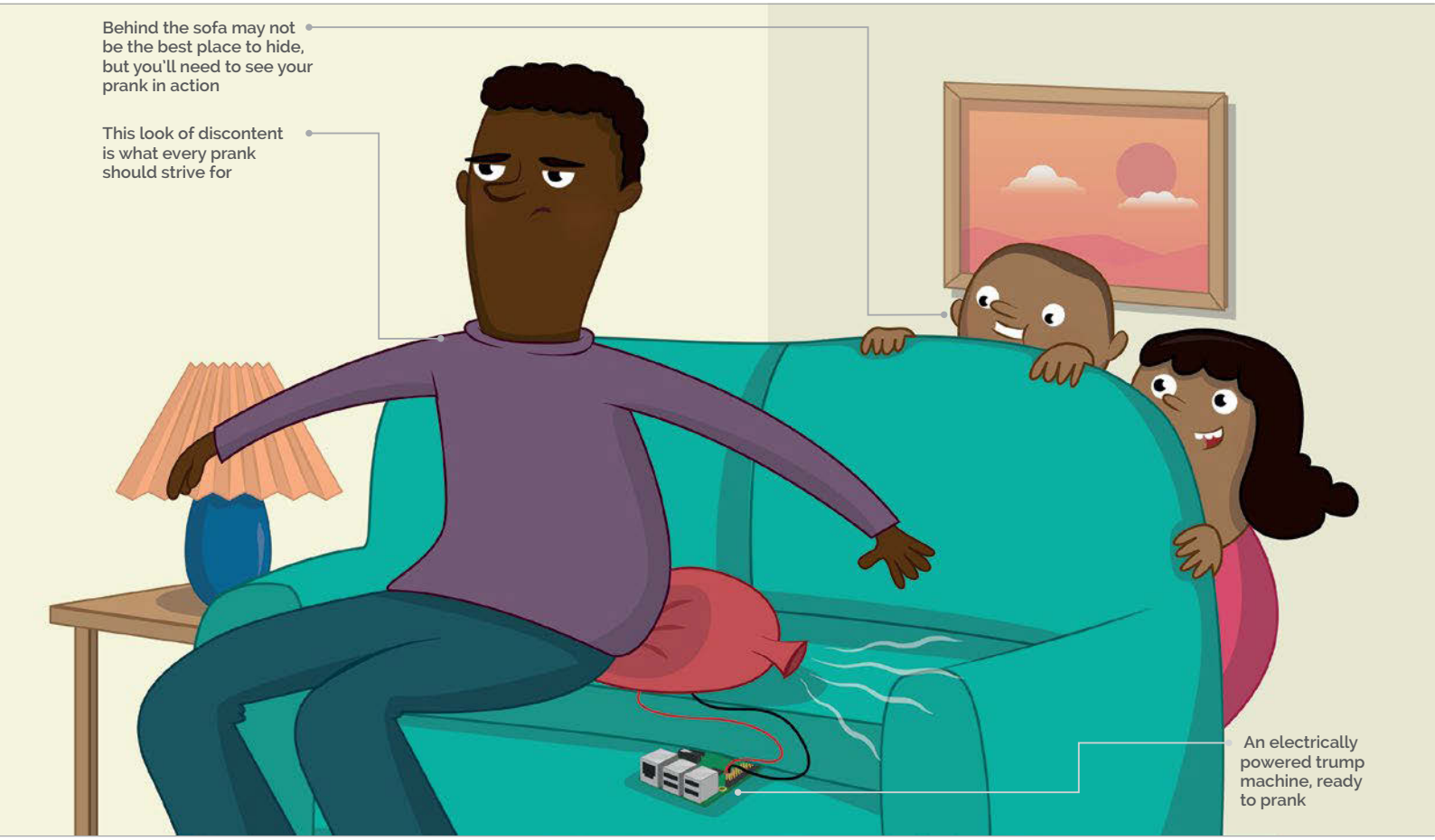
You'll Need

- 2× crocodile clips
- 2× male-to-female jumper leads
- A speaker
- Craft stuff: paper plates, cardboard, aluminium foil, glue, scissors, sponge, bulldog or paper clips, sticky tape

MAKE A WHOOPI CUSHION

A Pi-powered retro prank, perfect to get your mind thinking about a Pioneers project!

In the bad old days before TV and computers, the most popular form of family entertainment was the whoopee cushion, a tooty balloon originally made from a pig's bladder. The whoopee cushion would be inflated and hidden under Grandad's chair cushion. When he sat down, it would make a loud 'PARP!' noise, causing him to jump into the air and making his false teeth fly out. It was the best thing ever. This project brings the whoopee cushion up to date: there's no bladder and no need to blow it up. Thanks to the Raspberry Pi, you can also add whatever noises you want!


Behind the sofa may not be the best place to hide, but you'll need to see your prank in action
This look of discontent is what every prank should strive for
An electrically powered trump machine, ready to prank

Making the Whoopi cushion

Cut a circle from the cardboard to fit the centre of the 'eating' side of each paper plate. Tape or stick squares of foil onto the cardboard. These are your contacts: when they touch, they will complete a circuit. Using some copper tape, connect the foil to the edge of the plate. Chop the sponge up into cuboid chunks and glue them around the foil on one of the plates: this will stop the foil squares touching each other until someone sits on the plate. It should look something like **Fig 1**.

Put the two plates together so that the foil pieces are on the inside and facing each other. Offset the paper plates so that the copper tape sections are not touching.

You now have a 'cushion' made of two plates. You can tape the plates together, or you might choose to use bulldog or paper clips so that you can test that your cushion works correctly, and easily debug any hardware problems.

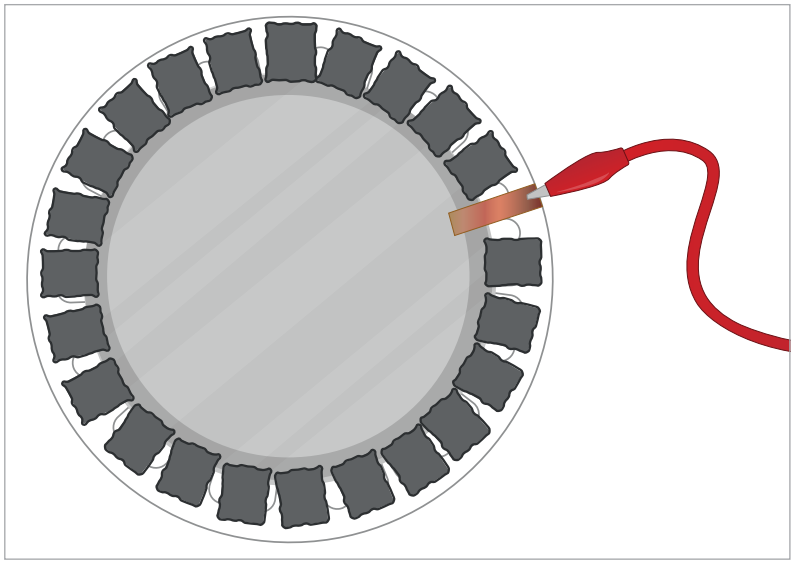


Fig 1 It looks a bit like some futuristic tech. We'll use it to make farting sounds

Connect the Whoopi cushion to the Pi

Plug one header lead (it doesn't matter which) onto a ground (GND) pin on the Pi (refer to **Fig 2**). Plug the other wire onto GPIO pin 2 (**Fig 3**, overleaf) and then clip a crocodile clip cable to one of the paper plates' copper tape sections. Connect the other end of the crocodile clip cable to the male pin of one of the connected jumper wires (**Fig 4**). Repeat these last two steps to connect the other plate to your Raspberry Pi. Your setup should look similar to **Fig 5**.

Test the sound

That's the hardware complete. Now for the software! We're going to use Python. Don't worry if you haven't used it before: just follow the instructions and you'll pick it up. You'll be using the command line to enter commands. To do this you'll need to open a terminal window by clicking on the terminal icon: it looks like a computer screen, and is found three icons along from the Menu button on your desktop.

Connect the speaker to the Raspberry Pi using the sound jack. Create a new folder called **whoopee** by typing the following command in the terminal and pressing Enter on the keyboard:

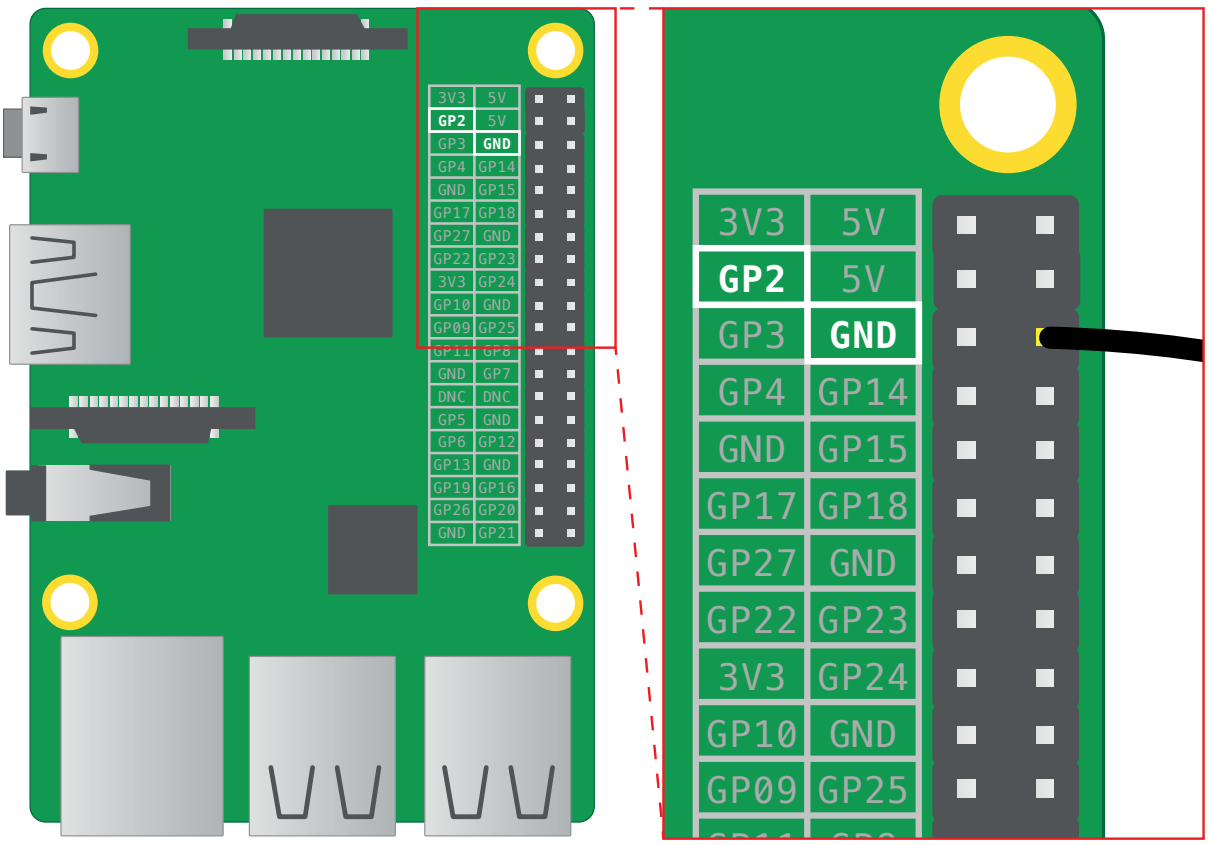


Fig 2 If you have an older Raspberry Pi model you'll only have 26 pins, but they have the same layout

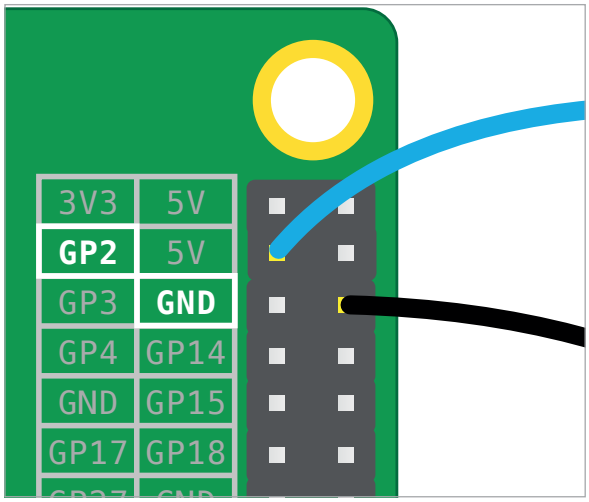


Fig 3 Again, this will be the same pin on any Raspberry Pi

```
mkdir whoopee
```

Next, use **cd whoopee** to enter the folder you have just created. We're going to need a sample sound file for this project, so we'll use one from Sonic Pi. Download this burp sample using the following command:

```
wget http://rpf.io/burp -O burp.wav
```

Now, test that you can play the sound file using **aplay** by typing:

```
aplay burp.wav
```

You should hear it from the speakers or headphones connected to your Pi. If you can't hear anything, make sure that your speakers are connected correctly. If this still doesn't work, you'll need to change your audio configuration.

MAKE YOUR OWN PRANKS AND WIN!

Think you can do better than this? The Raspberry Pi Foundation is running a competition for folks between 12 and 15, and all you need to do is make them laugh using digital technology. Sound easy? Sound fun? Then check out the website for more details: **magpi.cc/2iHKIP5**

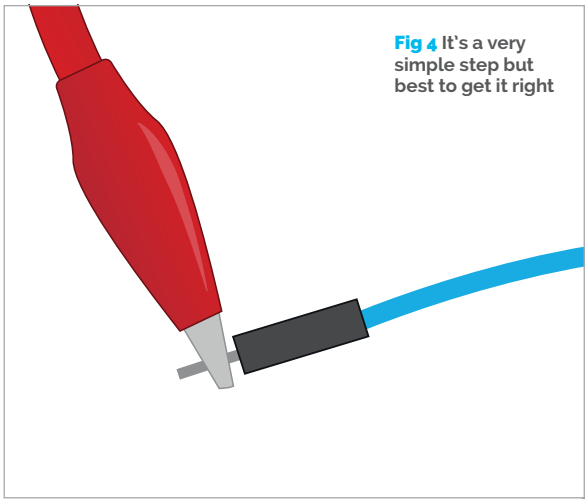


Fig 4 It's a very simple step but best to get it right

To switch audio to the headphone jack, return to the terminal window and type the following command:

```
amixer cset numid=3 1
```

If your Raspberry Pi is connected to the internet, you could search for some suitable trumping sounds. They need to be in WAV format to work. Alternatively, download our example sounds here: **rpf.io/farts**.

Write a program in Python

Open Python 3 (IDLE) from the Programming menu and click on File > New Window. This will open a blank file. Click on File > Save As, name the file **whoopee.py** and then type the following code into your file:

```
import os
import random
from time import sleep
from gpiozero import Button
```

This part of the code pulls in all the libraries that you're going to use to write your program. You'll then need to use the Button class in your code. You'll have to tell it that the button is on pin 2. To do this, write the following code in your new file:

```
button = Button(2)
```

Now create a list of all your sound effects and store them inside a variable that you can call later on in your code:

```
trumps = ['ben-fart.wav', 'ca-fart.wav',
'marc-fart.wav']
```

In Python, square brackets are used to create a list. Each item in the list is separated by a comma.

Once all the setup needed in the code is complete, you can move on to writing the part of the program that will make something happen when the button is

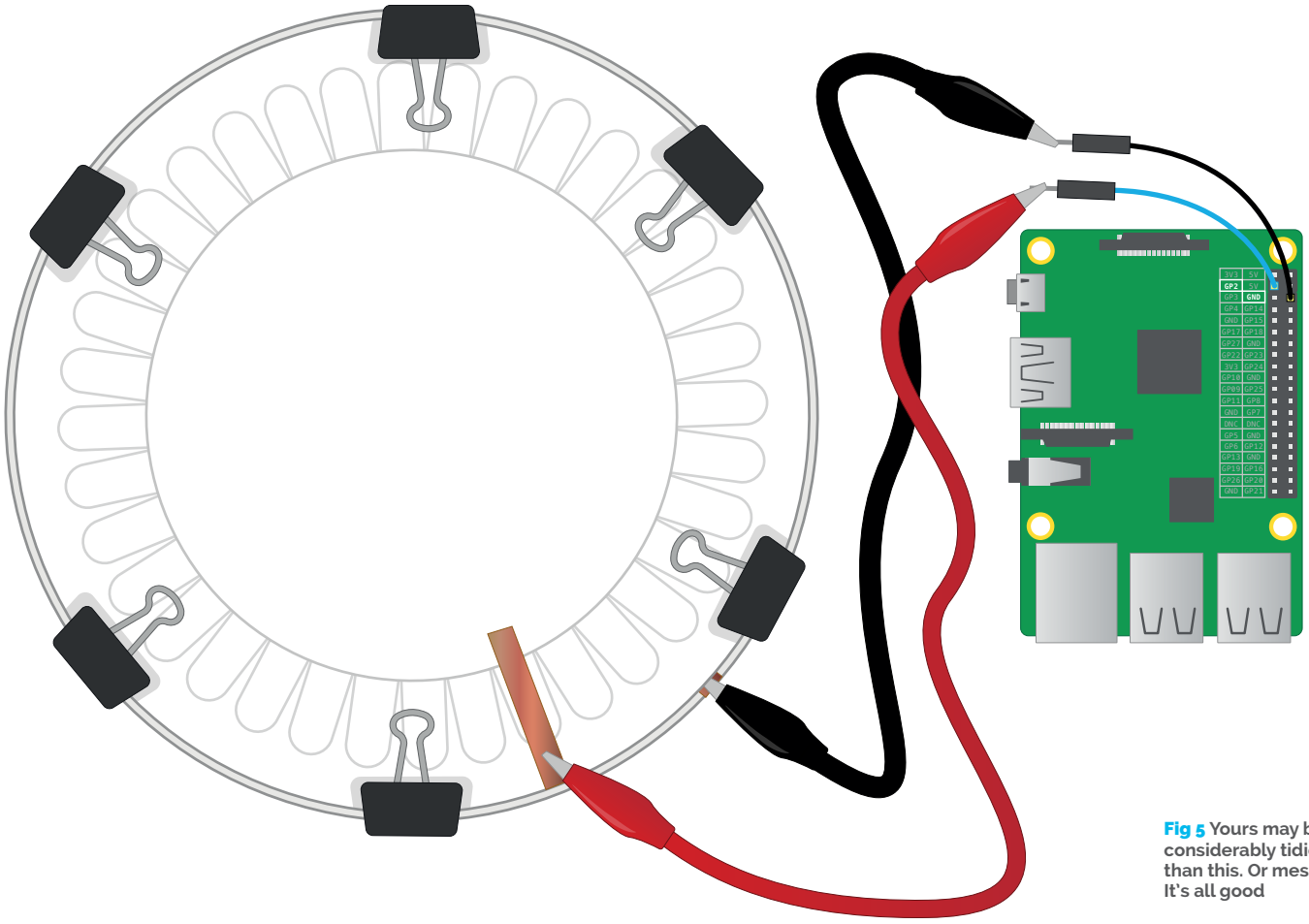


Fig 5 Yours may be considerably tidier than this. Or messier. It's all good

pressed. Begin by creating a loop using **`while True:`**. Then, add **`button.wait_for_press()`** inside the loop by indenting by four spaces. Each time around the loop, the computer waits for the button to be pressed.

On the next line, use the **`random.choice`** function to select a sound at random from the list you created earlier. That selected sound needs to be stored inside another variable which you can call **`parp`**! Type **`parp = random.choice(trumps)`**.

The next line will play the sound selected at random using **`aplay`**, as used earlier when testing sounds. Type **`os.system("aplay {0}".format(parp))`**. Finally, add **`sleep(2)`** to pause the program before it starts the loop again. Your code should look like this:

```
while True:
  button.wait_for_press()
  parp = random.choice(trumps)
  os.system("aplay {0}".format(parp))
  sleep(2)
```

Save the file by clicking on File > Save. Test that your code works by clicking on Run > Run Module. Use your hand to push the top plate of your Whoopi cushion down to make a connection between the foil sheets, and you should hear a fun sound. If it doesn't work first time, don't worry. Check your code through. Have you typed your code out exactly as you see it here?

whoopee.py

Language
>PYTHON 3

DOWNLOAD:
magpi.cc/WhoopiCushion

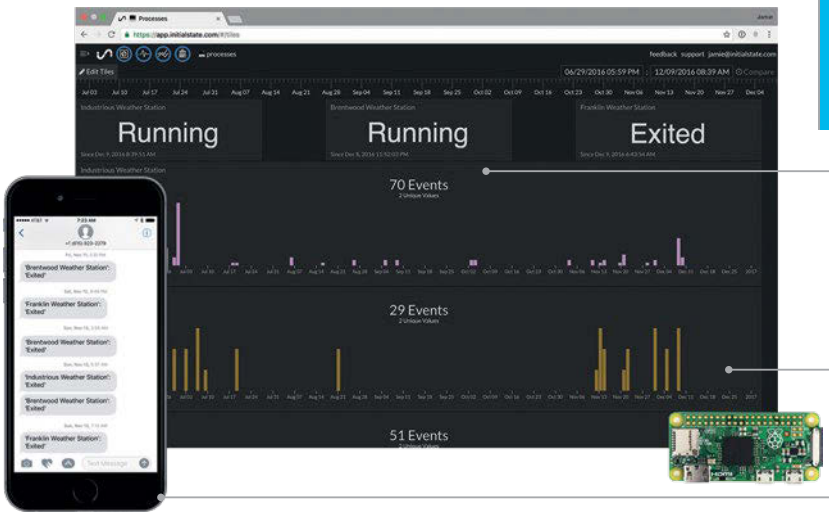
```
import os
import random
from time import sleep
from gpiozero import Button

button = Button(2)
trumps = ['ben-fart.wav', 'ca-fart.wav', 'marc-fart.wav']

while True:
    button.wait_for_press()
    parp = random.choice(trumps)
    os.system("aplay {0}".format(parp))
    sleep(2)
```

Setting it up

Carefully place your cushion in a spot where your victim will sit on it (obviously!), but not under a really heavy cushion where it will be squashed straight away. The tricky bit is setting up the Pi so that it can't be seen: remember, you'll need a plug socket to connect it to the power, unless you're using a battery pack. Run the program, and wait. Here's a hint: whistle tunelessly and look around at the ceiling. This will make you seem innocent, and will help to attract potential victims.

JAMIE BAILEY

Jamie is an electrical/system engineer, integrated circuit designer, and CEO/founder of Initial State, a data analytics service for Internet of Things devices. **initialstate.com**


Running
Running
Exited
70 Events
29 Events
51 Events
See the current status of each process at a glance
See a history of when processes exit to spot if there's any patterns
Receive a text message or email whenever a process exits

REMOTELY MONITOR YOUR PI PROCESSES

You'll Need

- Initial State account **magpi.cc/2kiYp77**

Create a web-based dashboard of all your important processes and get a notification when one exits

If you're using one or more Raspberry Pis to run a dedicated task (e.g. weather station or monitoring your beer fridge), you need those processes to run uninterrupted. A task that exits unexpectedly may need your immediate attention to avoid lost data, project delays, or a system failure. It's impractical, though, to manually babysit a bunch of Raspberry Pis. A better way to ensure continuous operation is to be alerted when a process exits, and be able to pull up a single dashboard to see the status of every important process running on every one of your deployed Raspberry Pis.

Initial State

We want to stream the status of each of our processes to a web service and have that service turn our data into a nice dashboard that we can access from our laptop or mobile device. Our data needs a destination, and we'll use Initial State as that destination. First, go to **magpi.cc/2kiYp77** and create a new account. Next, install the Initial State Python module onto your Pi:

```
cd ~
\curl -sSL https://get.initialstate.com/
python -o - | sudo bash
```

Follow the prompts, including the step of creating a test script to ensure you can successfully stream data from your Pi.

Monitor a process

Whenever you run a program or script on your Pi, you initiate a process. Each running process will have a unique process ID number (PID). You can see every process owned by you by typing **ps** at your Pi's command prompt. You will see a response that looks something like this:

PID TTY:	3359 pts/0	3373 pts/0
TIME CMD:	00:00:01 bash	00:00:00 ps

If you want to find every running Python process, you can use the following command:

```
$ ps -ef | grep py

root      2287     1  0 Dec09 ?
00:00:00 sudo python /home/pi/weather/
weather.py
```

In the example above, the PID of the weather Python script is 2287. Once we know the PID of the process we want to monitor, we can pass this number as a parameter into a simple Python script that will monitor the existence of this defined process and send either a 'Running' or 'Exited' response to our dashboard.

We will use the Python script located at **magpi.cc/2kj3BYs**. Before we can use this script,

Language
>PYTHON

DOWNLOAD:
magpi.cc/2iUSsAY

we need to set each of our user settings at the top of the file.

BUCKET_KEY specifies the data bucket our script will stream into. Make sure you use the same BUCKET_KEY for each script that should send data into the same dashboard.

ACCESS_KEY is your unique Initial State account key. This key is used to send data into your account from the scripts. You can find your ACCESS_KEY in the Streaming Access Keys section of your Initial State account settings.

PROCESS_NAME is the name of the specific process you're monitoring that you want to display on your dashboard. For example, **`PROCESS_NAME = "Franklin Weather Station"`**.

You can run this script as follows:

```
python monitor_process.py <pid>
```

To run this in the background without interruption:

```
nohup python monitor_process.py <pid> &
```

Manually finding the PID is cumbersome. Let's make it a bit easier by creating another simple script that will find the PID automatically and launch our **monitor_process** script.

Launch a monitored process

We can use a simple Bash script to do three things: launch our important task/program/process, find the PID of the task/program/process we just launched, and pass the PID into our **monitor_process** script.

The Bash script that we'll modify and use is located at **magpi.cc/2kiS23q**.

To use this script, simply modify the two user settings at the top of the file:

PROCESS2RUN is the process you want to run and monitor. Specify the full path.

MONITOR_SCRIPT is the **monitor_process.py** script we just created. Again, specify the full path.

Before you run this script, you'll need to make it an executable file. Run the following:

```
chmod u+x launch_process.sh
```

To run the script:

```
./launch_process.sh
```

We could stop here and have a pretty good solution for monitoring our processes. However, we need to consider what happens when we have a power outage.

Monitor a reboot

If power drops, our process will instantly die, as will our monitoring script and network connection. On reboot, we need to do two things: send an update to our process dashboard that our process exited, and automatically start our process and process monitor after the reboot completes.

First, let's create a script that will run on reboot to send a 'process exited' status to our dashboard. We'll use the script at **magpi.cc/2kiTfIa**. The user settings at the top of this script need to be set just like the other scripts. Make sure you use the same **BUCKET_KEY**, **ACCESS_KEY**, and **PROCESS_NAME** as in the 'Monitor a process' step. Once this script is set up, we need to call it automatically on reboot.

Automatically launch on boot

To automatically launch our scripts on boot, we will use **`crontab`**:

```
crontab -e
```

At the end of the file that opens, add the **monitor_reboot** script and the **launch_process** script. Specify the full path. For example, you'll add a single line that looks something like the following (with your specific paths):

```
@reboot python /home/pi/weather/monitor_
reboot.py && /home/pi/weather/launch_
process.sh
```

Notice the **monitor_reboot** script is followed by **&&**. This specifies that the **monitor_reboot** script should complete before the next command runs. If there's a power outage, it may take a few minutes for your network connection to be restored. After the delays specified in the **monitor_reboot** script expire, status updates will be sent to your dashboard and your process will be launched.

Once this is in place, everything will be automated. Every time your Pi boots, your process dashboard will update, and your process will launch and be monitored.

Dashboard and notifications

Each process that you monitor will add a data tile to your dashboard. This can be extremely useful if you have an application where multiple Pis are deployed across multiple locations. For example, if a Pi is used to monitor temperature in every room in your office building, you can quickly see the status of every Pi at a glance. You can enable SMS and/or email notifications for each process status by going to the Triggers section of your data bucket settings.

INITIAL STATE

Initial State is an easy-to-use platform for collecting data from connected devices and turning that data into dashboards, waveforms, notifications, and more. **initialstate.com**

FREQUENTLY ASKED QUESTIONS

Your technical hardware and software problems solved...

NEED A PROBLEM SOLVED?

Email **magpi@raspberrypi.org** or find us on **raspberrypi.org/forums** to feature in a future issue.

RASPBIAN INTERFACES

WHAT INTERFACES CAN I USE ON RASPBIAN?

PIXEL

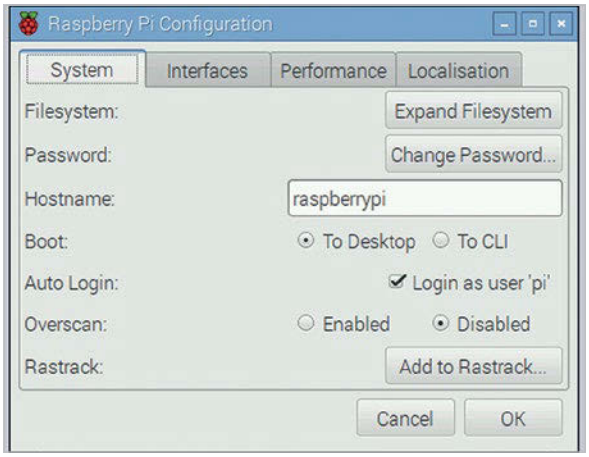
The graphical user interface (GUI) for Raspbian is the main way to use the Raspberry Pi. From here you can access traditional browsers, office suite apps, and the programming IDEs. This is the interface most people will use and the one which is recommended.

Command line

The classic command-line interface (CLI) requires you to input text commands and there's very little in the way of graphics outside of ASCII art. You can control the Raspberry Pi a bit more directly from here, while still being able to code.

Custom interface

You can also create custom interfaces for your Raspberry Pi: we've seen Python-powered phone-style interfaces you can use, and there are a few web-based interfaces around as well. The latter are very popular in CCTV or robotics applications.



Right Raspberry Pi Configuration holds all the important Raspbian settings

HOW DO I SWITCH INTERFACE?

Default interface

You can select which interface to use by default from the Raspberry Pi Configuration tool under Preferences in the program menu. It is set to boot to the desktop by default, but you can also set it to boot straight into the command line if it suits your usage.

Switch while running

You can switch between the desktop and CLI while the Pi running. From the command line, using the command `startx` will boot up the desktop. From the desktop, you can quickly switch to a command-line interface using **CTRL+ALT+F2**. You can then switch back to the desktop interface with **CTRL+ALT+F1**.

Custom interfaces

Most custom interfaces, at least ones running directly from the Pi, require the desktop be running so it can use the graphical tools to display them. Keep that in mind while making something that turns on automatically at boot. Check what is required for web interfaces.

AUTO LOGIN

Defaults

By default, the Raspberry Pi will log you in whichever way you boot up. For most uses this is fine; however there will always be cases where this is quite insecure. As education is the main focus of the Raspberry Pi Foundation, though, this is why it's turned on.

Change it

You can change whether or not the Pi automatically logs in. Go to the Raspberry Pi Configuration menu. Underneath the option to change which desktop it boots into, you'll find the option to turn auto login on and off.

Password

For Raspbian, the default user name is **pi** and the password is **raspberry**, so you'll need to remember that if you turn off auto login. If you're turning it off for security reasons, it's a good idea to change the password as well. Find out how to do so here: **magpi.cc/2iqm9pO**.

FROM THE RASPBERRY PI FAQ
RASPBERRYPI.ORG/HELP

What size SD card do I need for the Raspberry Pi?
Whether you want to use the NOOBS installer or a standalone distro image, the minimum size SD card we recommend is 8GB. This will give you the free space you need to install extra packages or make programs of your own. The original Raspberry Pi Model A and Model B require full-size SD cards. The newer Raspberry Pi Model A+, Pi Model B+, Pi 2 Model B, Pi Zero, and Pi 3 Model B require microSD cards.

What size SD cards can it support?
We have tried cards up to 32GB, and most cards seem to work fine. You can also attach a USB stick or USB hard drive to provide extra storage.

What happens if I brick the device?
You can restore the device by reflashing the SD card.

Will it run Android?
No. While a version of Android can be found in the forum, it is not stable enough for everyday use. There are no plans to continue working on it, since Android does not provide any enhancement to educational purposes that are not already fulfilled more readily with existing software: we see it as a platform for consumption, not for creation.

Will it run [insert name of program here]?
In general, you need to look to see whether the program you want can be compiled for the ARMv6 (Pi 1) or ARMv7 (Pi 2/3) architecture. In most cases, the answer will be yes. Specific programs are discussed on our forum, so you might want to look there for an answer. Ultimately, though, nothing beats grabbing a Raspberry Pi and finding out the answer through direct testing!

WD PIDRIVE NODE ZERO

COMPACT AND PRE-ASSEMBLED WITH A HDD AND RASPBERRY PI ZERO.

Build your own mini DLNA + Samba file server, wireless mobile storage device, or a Pi MusicBox with local storage, and more.

Includes:
- 314GB WD PiDrive, Raspberry Pi Zero, and custom adapter board
- SD card with starter software
- Mini HDMI adapter cable

ONLY $44.95

WDLABS

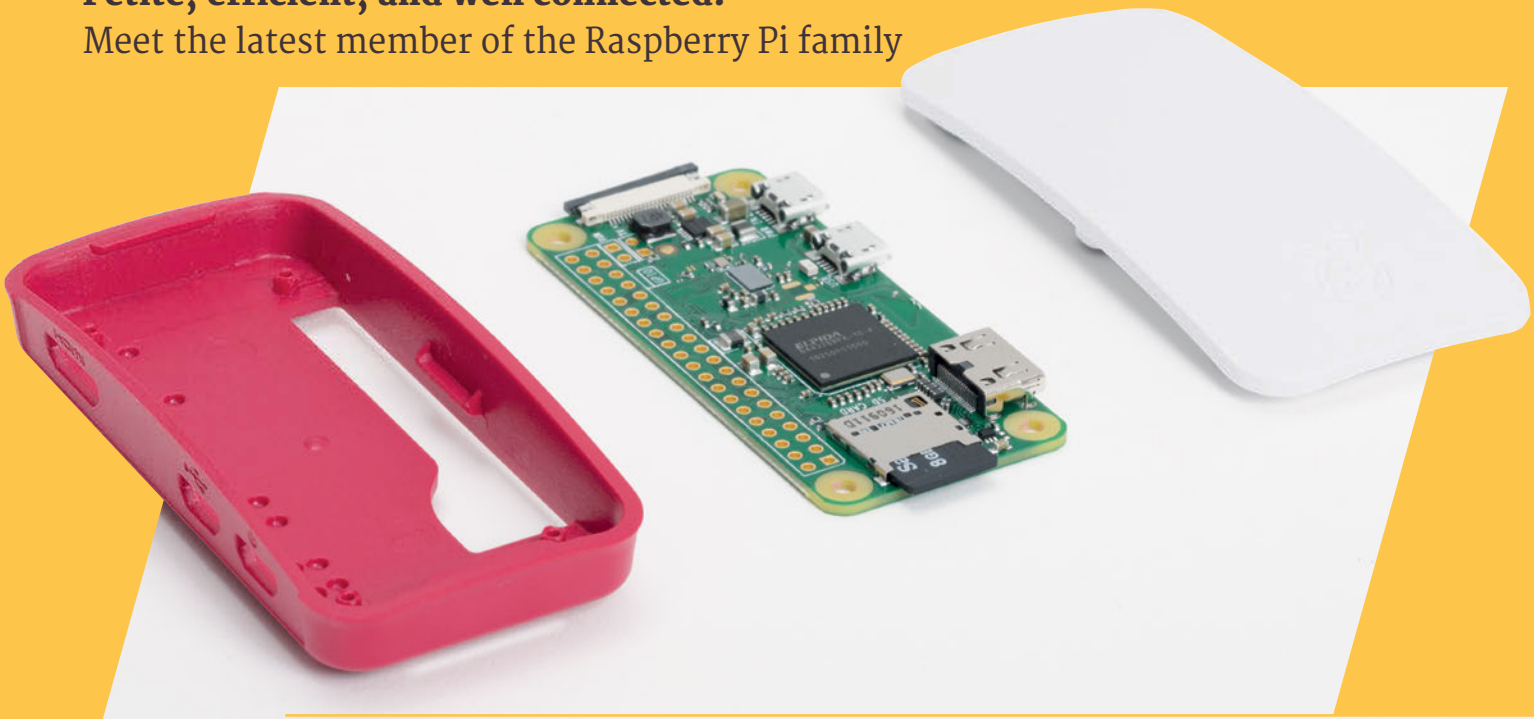
Learn more at **wdlabs.wd.com/magpi55b**

Western Digital, WD, the WD logo, WDLabs, and the WDLabs logo are registered trademarks or trademarks of Western Digital Corporation or its affiliates in the U.S. and/or other countries. Raspberry Pi is a trademark of the Raspberry Pi Foundation. microSD is a trademark of SD-3C, LLC. The term HDMI is a trademark or registered trademark of HDMI Licensing LLC in the United States and other countries. All other marks are the property of their respective owners. As used for storage capacity, one gigabyte = one billion bytes. Total accessible capacity varies depending on operating environment. Product specifications subject to change without notice. Pictures shown may vary from actual product. © 2017 Western Digital Corporation or its affiliates.

PI ZERO W

Petite, efficient, and well connected.

Meet the latest member of the Raspberry Pi family



The big news this month is the introduction of a new computer model from the Raspberry Pi Foundation.

The Pi Zero W is the same small revolutionary board that rocked the maker community, but it now packs built-in wireless LAN networking and Bluetooth.

It also comes with an incredible new case. With switchable lids, this enables the Pi Zero W to function as a camera, provides access to its GPIO pins, or lets it act as a secure node.

A third of the size of a Raspberry Pi 3, the Pi Zero W still has enough power to run a full operating system like Raspbian.

This update to the Pi Zero has the same small form factor, but crams in wireless LAN networking too. With the Pi Zero W, you can connect directly to a local wireless network without having to attach a USB dongle.

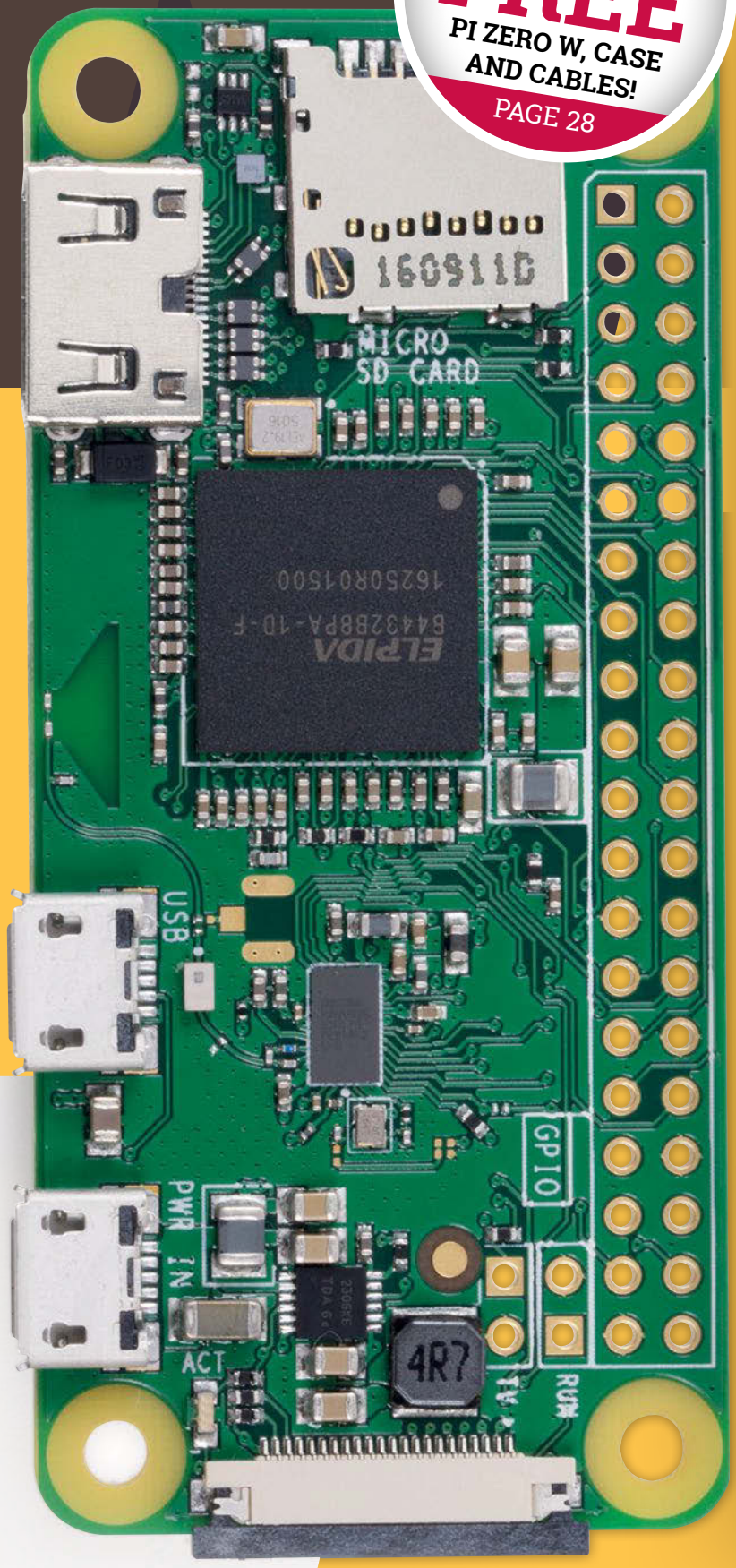
The Pi Zero W gets the same wireless networking capabilities first deployed on the Raspberry Pi 3.

Squeezing wireless networking onto a Pi Zero board hasn't been easy, and the way Raspberry Pi went about it has shown considerable ingenuity. The new Pi Zero case is also a thing of wonder.

But it's what you can now do with a Pi Zero W that is going to make a difference.

The Pi Zero W can connect automatically to a local network. With its small size and low-energy footprint, it's ideal as a node for projects around the home, as well as IoT and wearable projects.

SUBSCRIBE TODAY AND GET A FREE PI ZERO W, CASE AND CABLES!
PAGE 28



PI ZERO W

Technical Specifications

Dimensions:
65mm × 30mm × 5mm

SoC:
Broadcom BCM2835

CPU:
ARM11 running at 1GHz

RAM:
512MB

Wireless:
2.4GHz 802.11n wireless LAN

Bluetooth:
Bluetooth 4.1 LE

Power:
5V, supplied via micro USB connector

Video & Audio:
1080P HD video & stereo audio via mini-HDMI connector

Storage:
MicroSD card

Output:
Micro USB

GPIO:
40-pin GPIO, unpopulated

Pins:
Run mode, unpopulated; RCA composite, unpopulated

Camera Serial Interface (CSI)

Price:

US	UK	EU
$10	£9.60	€11
EX-TAX	INC VAT	INC VAT

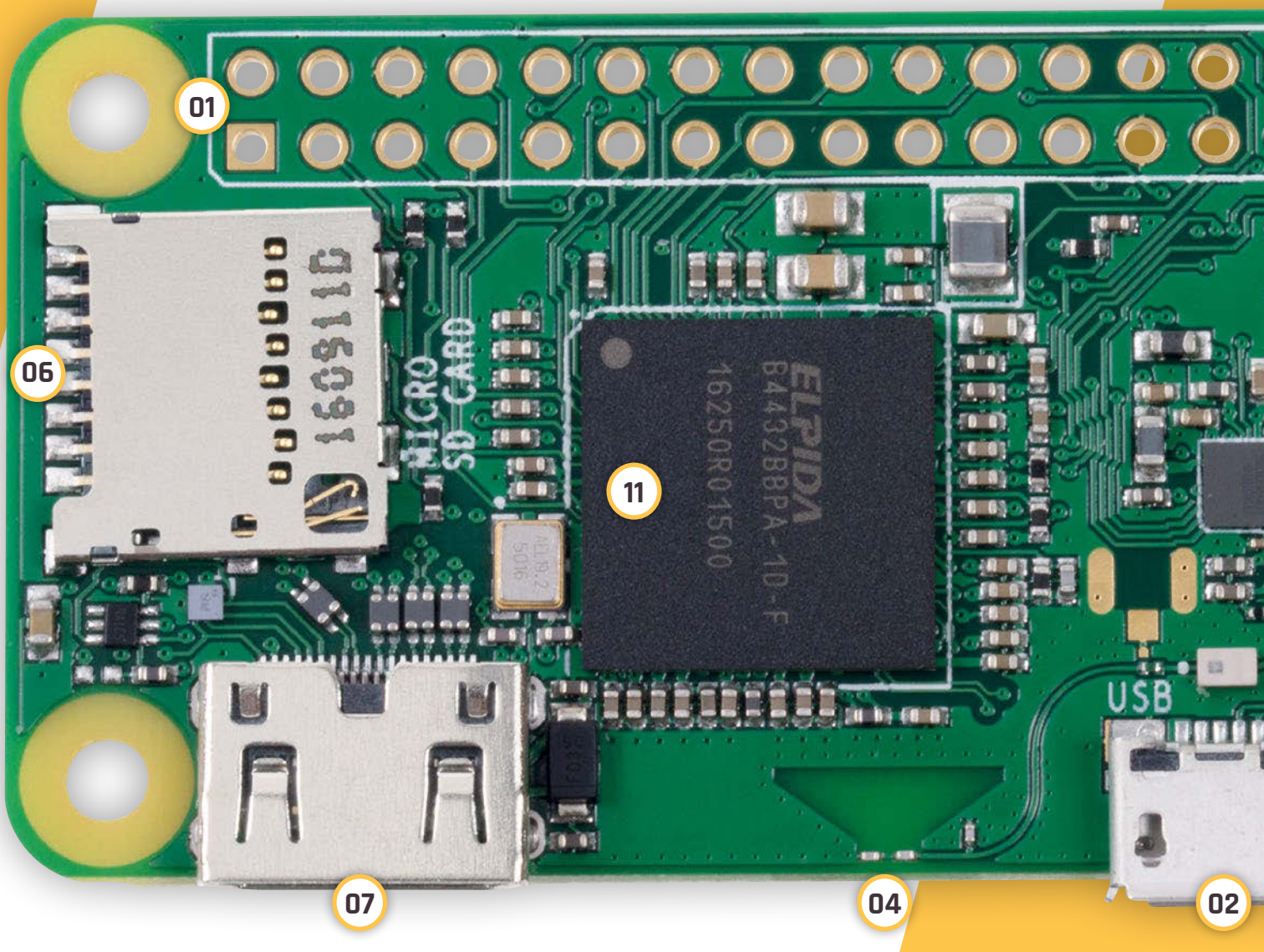
THE BOARD

Up close with the technology packed onto a Pi Zero W board

The technology found in the Pi Zero W is similar to that of the original Pi Zero, only now everything has had to make space for an incredibly smart radio antenna. This design enables the minute Pi Zero board to pack a complete wireless LAN and Bluetooth connection module.

The Pi Zero W contains all of the technology from the original Pi Zero, but with a slightly redesigned board.

The technical surge forward is thanks to the presence of 2.4GHz 802.11n wireless LAN and Bluetooth



4.1 LE. These make the Pi Zero W much more accessible to makers.

It reminded us how packed with technology the Pi Zero is. It features a full 40-pin GPIO header (unpopulated, so you will need to solder the pins on yourself).

Two extra TV pins are used to connect the Pi Zero W to an RCA television, while two pins marked RUN can be used to plug in a reset button.

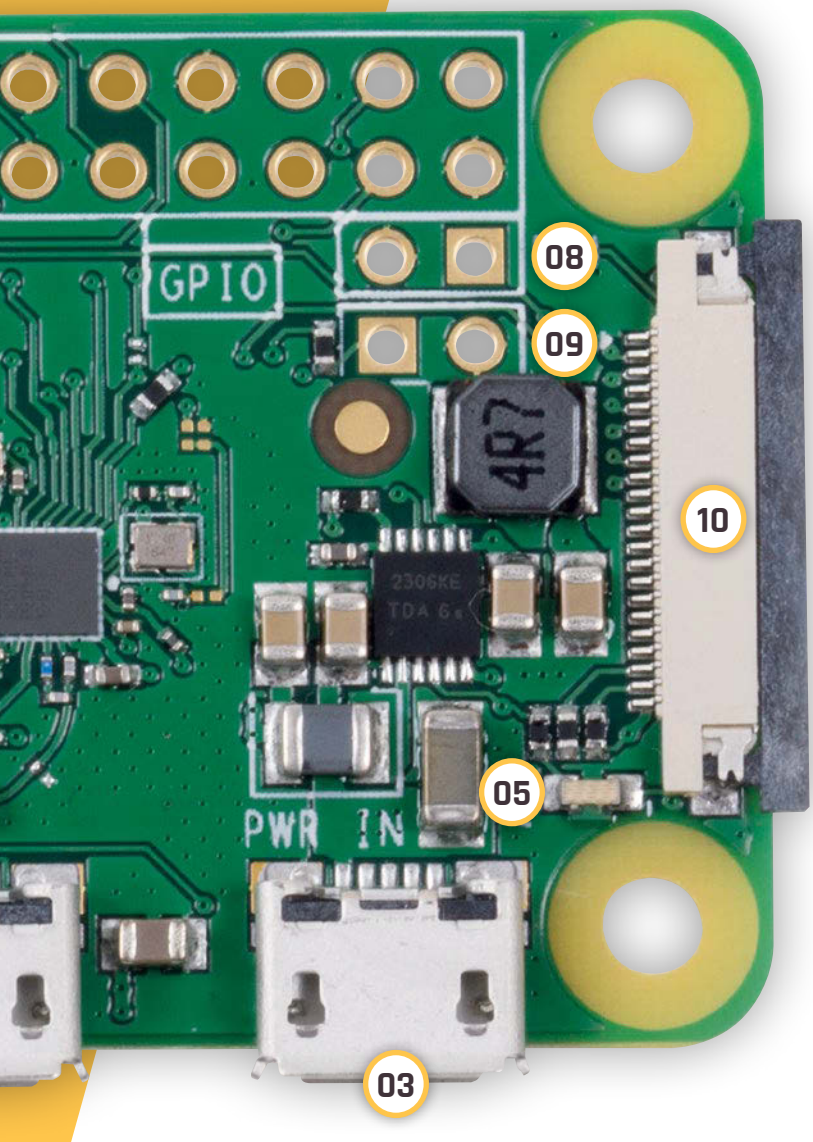
A single USB On-The-Go connection allows you to hook up devices to the Pi Zero W, and the mini HDMI socket connects the board to monitors and TVs.

To hook up stock cables, you'll want a Raspberry Pi Zero Adapter Kit (£4, **magpi.cc/2kPZNzn**) to get the most from your Pi Zero W.

A CSI connection supports the Camera Module v2 (**magpi.cc/28IjIsz**). This adds a Sony IMX219 8-megapixel camera to the Pi Zero W. It's especially pertinent thanks to the new case.

The processor remains a Broadcom BCM2835 containing an ARM11 running up to 1GHz. The Pi Zero W is a full computer, capable of running Raspbian and other Linux operating systems.

> "The Pi Zero W is a full computer, capable of running Raspbian and other Linux operating systems"



01. GPIO header
Pi Zero W has the same 40-pin GPIO header as the larger Raspberry Pi models, but it's unpopulated. Adding pins to the Pi Zero W board is a great first soldering project.

02. Micro USB
A single micro USB port is used to provide connectivity. Only now you don't need to add a USB WiFi dongle, so it'll be easier to connect just a keyboard and mouse.

03. Micro USB power
The second micro USB port is used to provide power to the board.

04. Wireless LAN
This triangular shape is used to boost the wireless connection. The signal bounces around the triangle and connects to the points at the bottom (before being fed up the pipes to the side).

05. Bluetooth 4.1
The Pi Zero also connects to Bluetooth 4.1. This connectivity makes it easy to hook up wireless devices and share files with the board.

06. MicroSD card
A microSD card is used to provide the storage for the Pi Zero W.

07. Mini HDMI
A mini HDMI connection is used to connect the Pi Zero W to a TV or monitor.

08. RUN pins
The two RUN pins can be used to wire up a reset button.

09. TV pins
The TV pins can be used to add an RCA video connection. Perfect for hooking up a Pi Zero W to older television sets (for retro gaming perfection).

10. Camera Module
A camera port was added to the Pi Zero 1.3 and remains on the Pi Zero W. The Camera Module is tightly integrated with the Pi Zero thanks to the new case.

11. Broadcom BCM2835
This Broadcom BCM2835 SoC is the heart of the Pi Zero W. It contains An ARM 11 CPU running at up to 1GHz.

BEHIND THE PI ZERO W

We chat with Roger Thornton, principal hardware engineer at the Raspberry Pi Foundation

Roger Thornton is the person in charge of the Raspberry Pi Zero W project: developing the new board has been a labour of love for him.

The first Raspberry Pi product to gain wireless networking was the Raspberry Pi 3 (released shortly after the original Zero). Developing this product gave Raspberry Pi the experience they needed to bring wireless networking to the Zero.

> "They're very clever boffins. It's a really neat design"

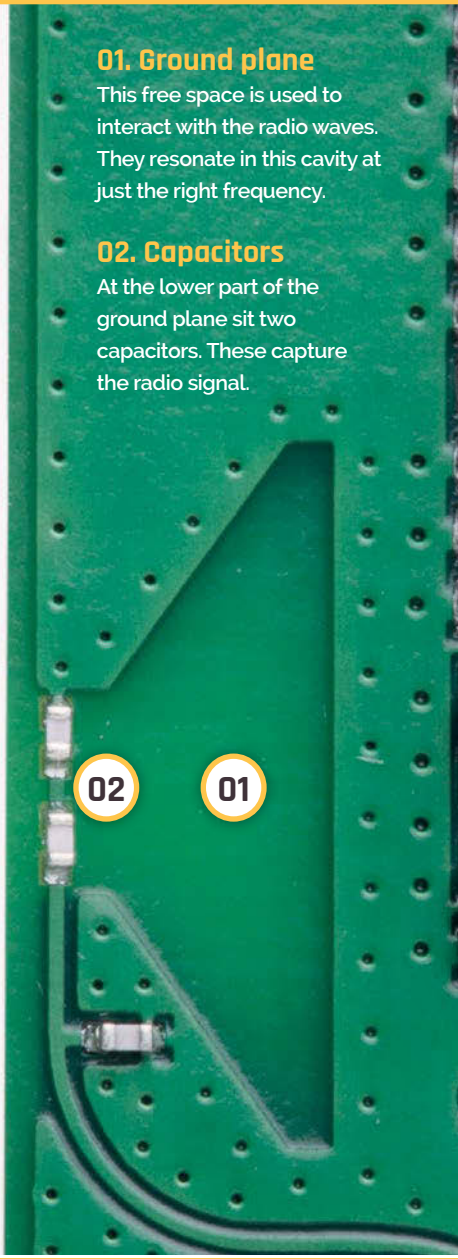
There was no easy way the wireless layout from the larger Raspberry Pi 3 board would fit on the diminutive Zero. "We thought it was impossible," muses Roger, "but we were getting better at squeezing features onto products, so we did a little playing around with the design." According to Roger, there was a lot of "pushing and shoving" to fit all the components into a restricted area, and onto one side of the board.

The Raspberry Pi 3 layout was based on a two-sided Broadcom reference design, with a 'chip' antenna. In contrast, the Zero W uses a single-sided layout with a PCB antenna; this is the trapezoidal shape between the mini HDMI and micro USB sockets on the bottom edge of the board.

"The Raspberry Pi 3 antenna is a surface-mount component," explains Roger, "whereas the Zero W antenna is a resonant cavity which is formed by etching away copper on each layer of the PCB structure."

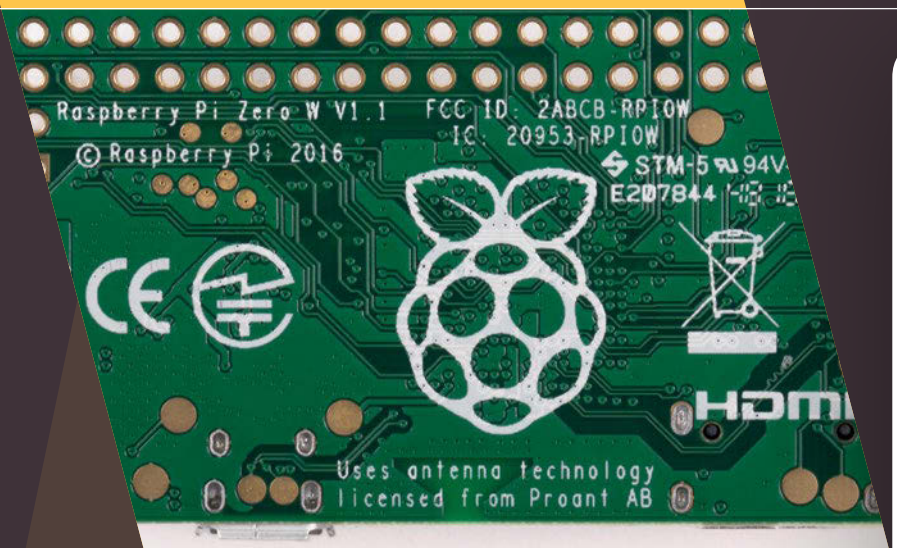
The technology is licensed from a Swedish company called Proant (you can see the credit on the reverse of the Zero W board).

"They're very clever boffins," says Roger. "It's a really neat design."



01. Ground plane
This free space is used to interact with the radio waves. They resonate in this cavity at just the right frequency.

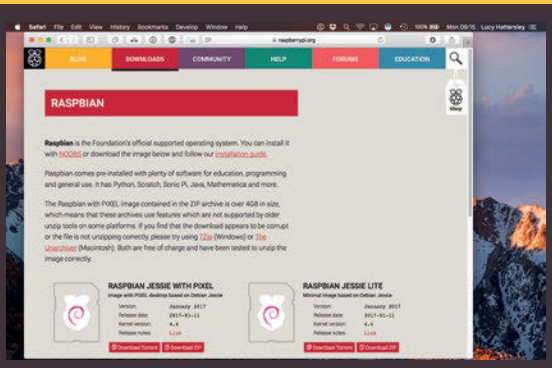
02. Capacitors
At the lower part of the ground plane sit two capacitors. These capture the radio signal.



The white text is added to the PCB using a silk-screen technique. These markings are used to provide information and branding for the board

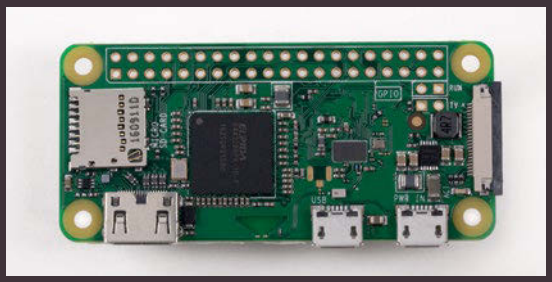
The intelligent layout enables all the circuitry to be placed on one side of the PCB, keeping the cost down

SET UP A PI ZERO W



>STEP 01
Download Raspbian

You'll need the latest version of Raspbian with PIXEL from **raspberrypi.org/downloads**. Make sure you get the most recent update, as the latest version adds Pi Zero W support. Put your Pi Zero W inside its case.



>STEP 02
Set up the Pi Zero W

Flash the microSD card and insert it into your Pi Zero W. Use a USB On-The-Go adaptor to connect your keyboard and mouse to the micro USB port marked 'USB' (chain the mouse via a keyboard USB if possible). Use a mini HDMI to HDMI cable (or adapter) to connect the Pi Zero W to a monitor.

>STEP 03
Power on

Connect a power supply to the micro USB port marked 'PWR'. The Pi Zero W will power up into Raspbian with PIXEL. Click on the wireless icon in the top-right of the screen to locate your local wireless network. Enter your wireless password to connect the Pi Zero W to the network.

PI ZERO W CASE

A whole new way to hold your Raspberry Pi

Launching alongside the Pi Zero W is a new case. The design is similar to the stunning Raspberry Pi 2/3 case. The same design team, Kinneir Dufort, is responsible for creating the Pi Zero W case.

It's an amazing case which shares a lot of design DNA with the original. It also comes in the same red and white colour scheme.

The case is composed of two parts: a red base and a white lid. They clip together to contain the Pi Zero. On the rear of the red case is a cut-out to allow room for GPIO pin edges and solder.

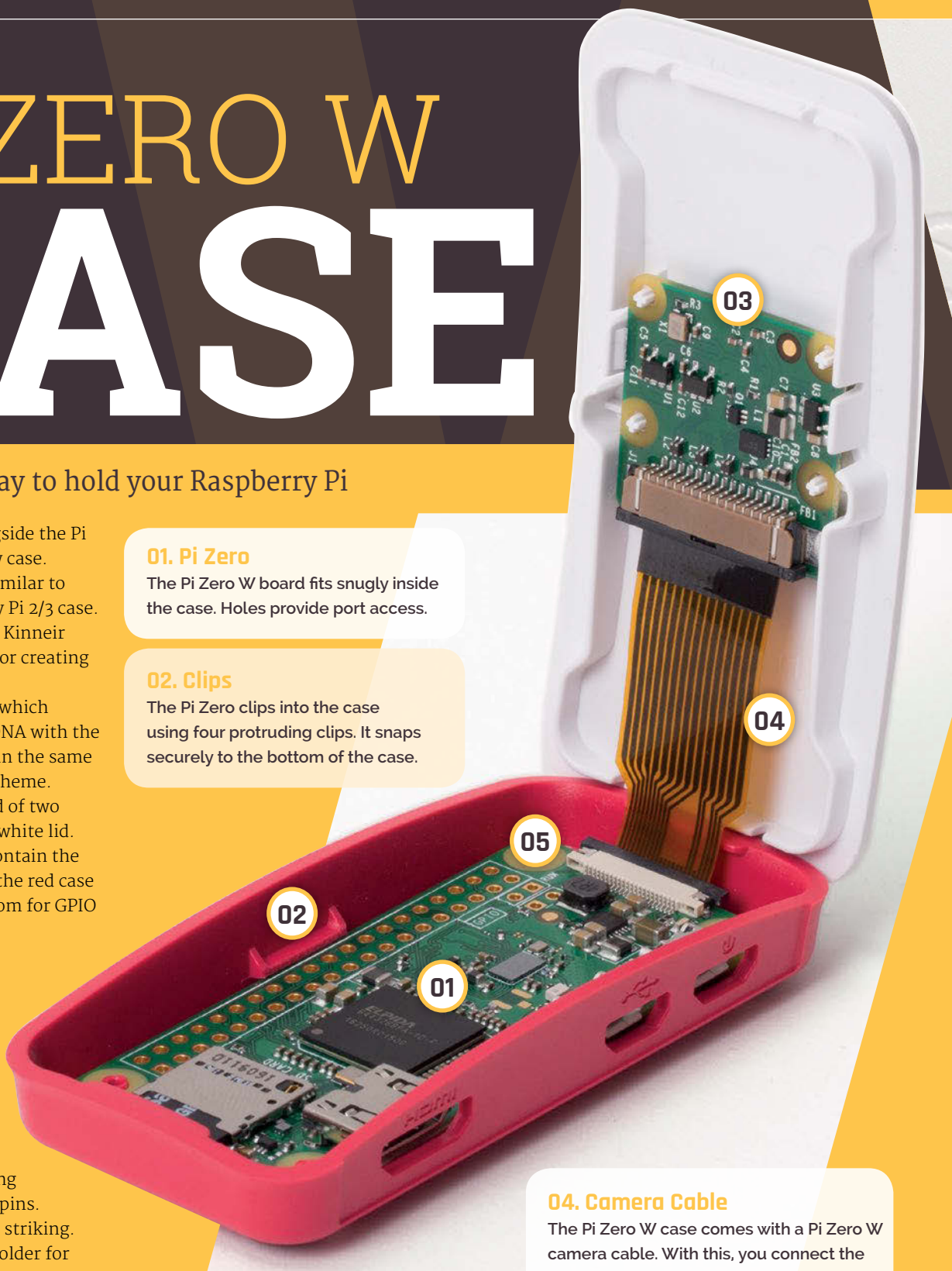
The white cover is interesting. You don't get just one lid; three are available...

The first is a solid cover that keeps your Pi Zero W components secure.

The second has a rectangular slot enabling you to access the GPIO pins.

The third is the most striking. This cover contains a holder for the Camera Module. The camera clips onto the inside of the lid, and a perfectly sized gap enables it to take photos through the cover.

Each Pi Zero W case comes with all three covers, so you can use it whichever way you choose.



01. Pi Zero

The Pi Zero W board fits snugly inside the case. Holes provide port access.

02. Clips

The Pi Zero clips into the case using four protruding clips. It snaps securely to the bottom of the case.

04. Camera Cable

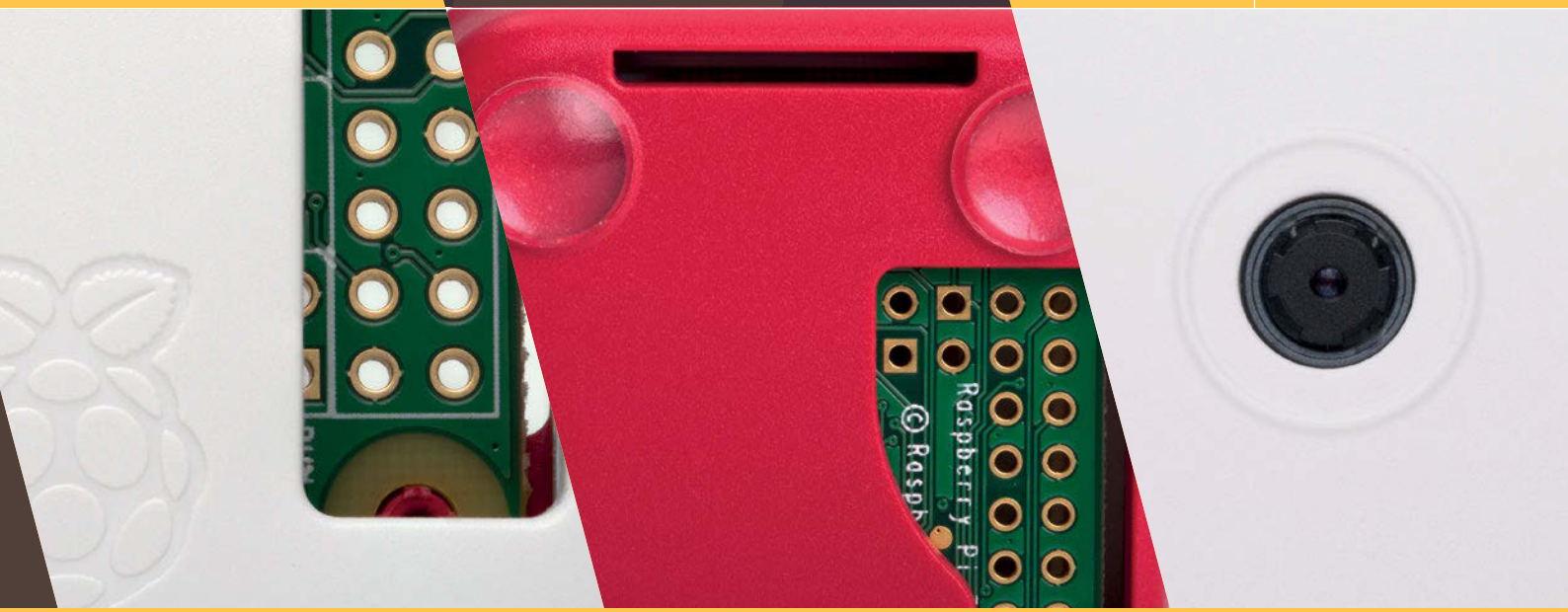
The Pi Zero W case comes with a Pi Zero W camera cable. With this, you connect the Camera Module to the Pi Zero W board.

03. Camera Module

The Camera Module is attached to the inside of the camera case lid. Like the Pi Zero W board, it snaps into place with the four plastic protrusions.

05. GPIO header

The case lids are interchangeable, and one lid enables direct access to the GPIO pins on the Pi Zero W.



THREE COVERS

Secure cover

The safe cover clips on to the Raspberry Pi case, completely sealing up the Raspberry Pi. Perfectly shaped holes on the side of the case enable you to access the USB and HDMI ports. It makes for the cutest computer around.

GPIO cover

The GPIO cover has a cut-out section that enables you to access the GPIO pins without removing the lid. Another cut-out in the base allows room for solder and pin edges (on the reverse of the board), but could be used for GPIO access too.

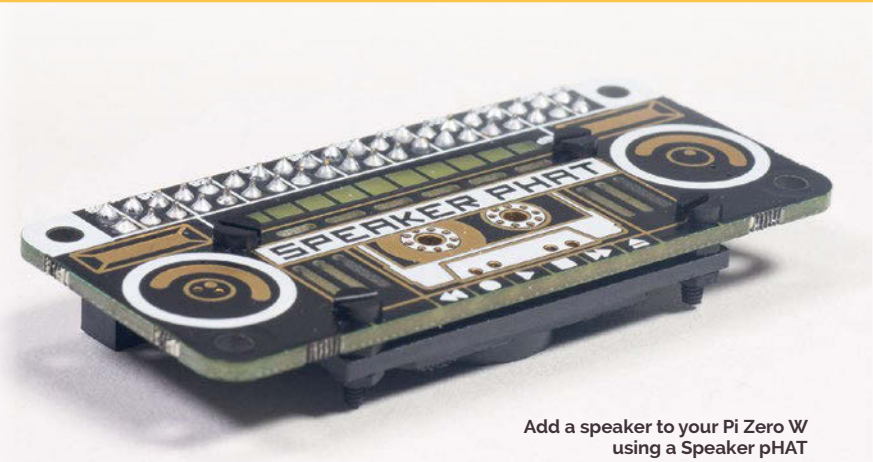
Camera cover

We think the camera cover is the star attraction. It holds the Camera Module in place and resembles a digital camera. We can't wait to see what Pi Zero W projects the community makes with integrated wireless and camera functionality.

PI ZERO W PROJECTS

Five great project ideas for you to build with your new Pi Zero W

AMAZON ECHO



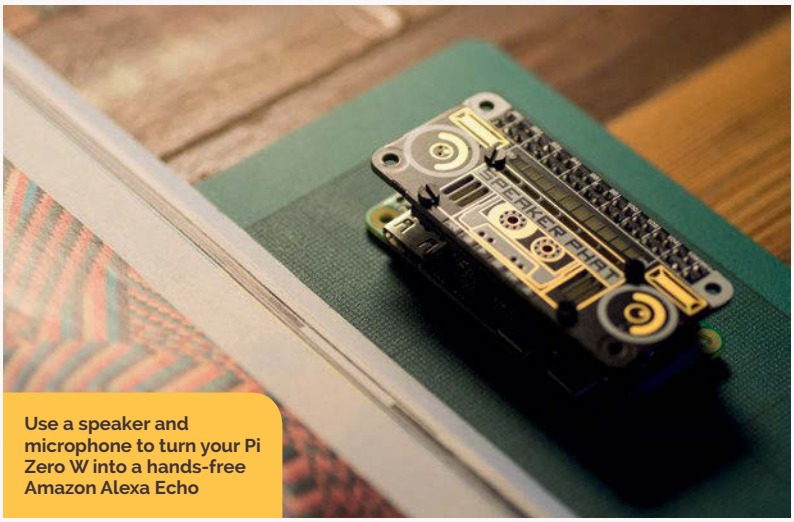
Add a speaker to your Pi Zero W using a Speaker pHAT

Amazon Alexa is an intelligent assistant making a lot of waves in the tech community. Amazon sells two Alexa products: the Amazon Echo and Echo Dot.

Amazon has also created a homebrew version of Alexa for the Raspberry Pi.

Now that the Pi Zero W has wireless LAN, it's a perfect portable Amazon Alexa device. All you need are some speakers and a USB microphone.

Pimoroni's Speaker pHAT (£12, **magpi.cc/2kXdZsE**) is a neat solution. Combine it with a USB microphone using the USB to USB On-The-Go adapter.



Use a speaker and microphone to turn your Pi Zero W into a hands-free Amazon Alexa Echo

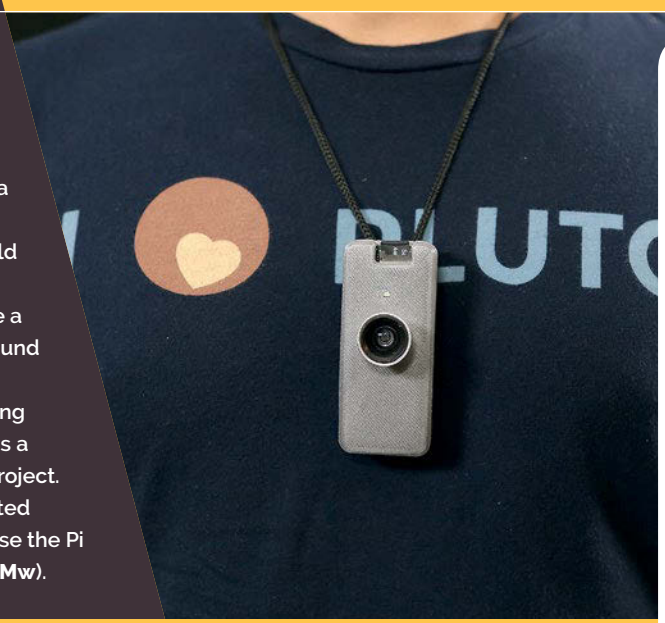
Previously, you also had to add a button to the Raspberry Pi version of Alexa, but a recent update has added an 'always on' mode to the Raspberry Pi build. Alexa now listens for a vocal cue, such as the word 'Alexa' and then starts monitoring. You can change the vocal signal to any name of your choosing; 'computer' is a popular choice for Star Trek fans.

A GitHub repository has all the information you need to set up Amazon Alexa on a Raspberry Pi (**magpi.cc/2kXrdW2**). There is a complete guide here (**magpi.cc/2kXfPcO**). Frederick Vandenbosch has written a guide to installing Alexa on a Raspberry Pi Zero (**magpi.cc/2kXkY4A**).

WEARABLE CAMERA

The Pi Zero W case with camera lid lends itself perfectly to making a mini camera. You could add a button to the case using Sugru to build a digicam, or use a Bluetooth camera button (as found on selfie sticks).

We also like the idea of making a wearable camera. Adafruit has a great tutorial on creating this project. Adafruit used a unique 3D-printed case, but now you can simply use the Pi Zero W case (**magpi.cc/2kXASMw**).



MOTION-SENSING CAMERA

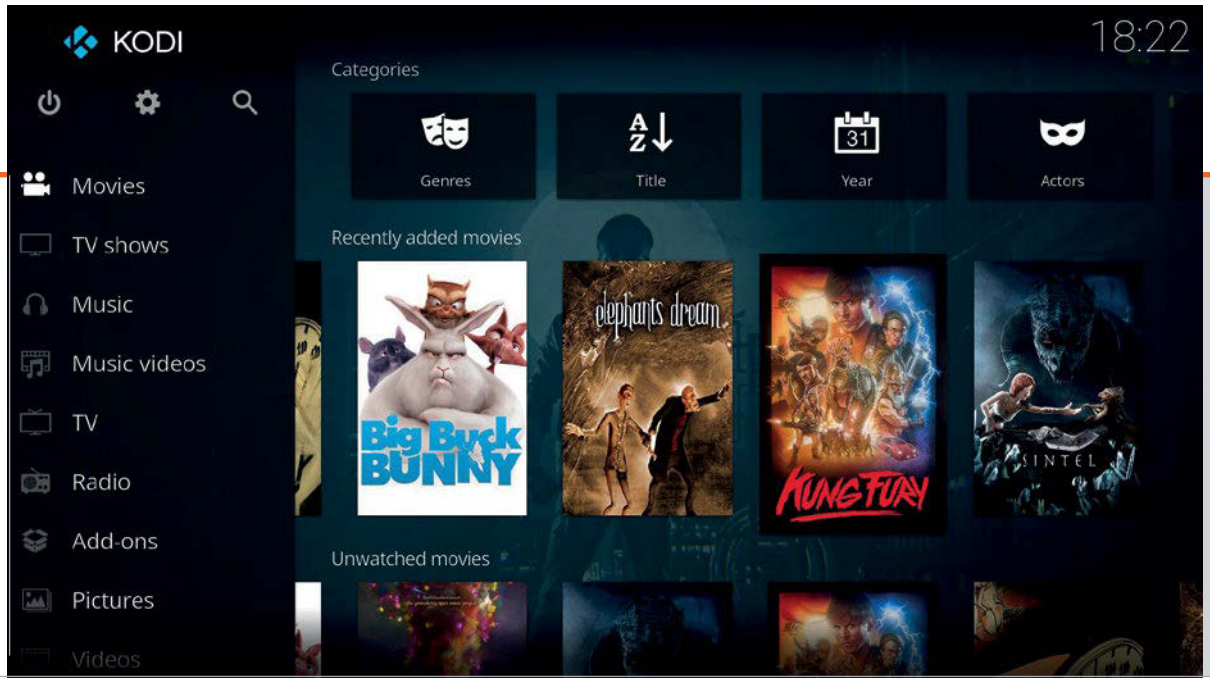
The low power draw of the Pi Zero W also makes it ideal for use as a security camera, or as an outdoor wildlife camera. Hook it up to a Zero LiPo (**magpi.cc/2kXI8YW**) and you'll get a steady stream of power. Connect the Pi Zero camera and use the wireless LAN to send alerts when it detects motion. Take a look at Mark West's project (**magpi.cc/2kXznhq**).

FILE SERVER

The low power draw of the Pi Zero W and its wireless connectivity make it ideal for use as a connected node. Add some storage and turn it into always-on network storage. This has always been a great Raspberry Pi project, but it is especially easy with the Zero W's built-in wireless connectivity.

SMART CAR DASH

The Pi Zero W is easy to mount on a car dashboard using a Zero View mount (**magpi.cc/2kXIsa2**). You can use it as a dash cam to record footage. With the wireless LAN, you can connect to a personal hotspot (from your mobile phone). Consider checking out the open-source Dride project for software (**magpi.cc/2lnBwjt**).



Maker Says

"An award-winning free and open-source software media centre"

Team Kodi

KODI 17.0 KRYPTON

The premier HTPC software gets an update with a new look and some new tech behind the scenes

Related

OSMC

OSMC runs a Kodi back end, but it uses its own interface and some other optimisations that the makers believe work better.

FREE

osmc.tv

A new Kodi release is always cause for a little celebration. It's one of the most successful, long-running pieces of open-source media software of any type, and it powers so many other popular media centres. The latest release of Kodi has some minor improvements behind the scenes, with an overhauled video playback engine which should benefit these other media centres in the long run. However, one of the biggest things that users of stock Kodi will notice is that there's a new default skin that replaces the venerable Confluence.

Bare-bones

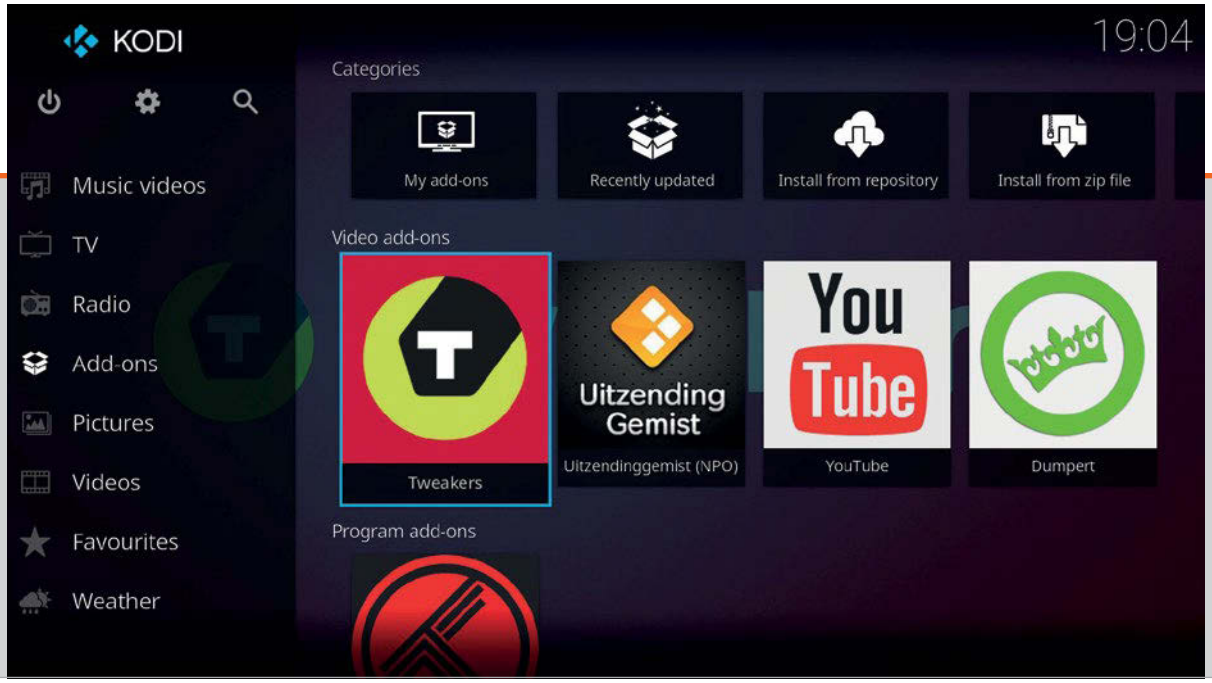
Skins and interfaces can be changed, though, so before we delve into that let's talk about the video playback, as it has had an update.

Long story short, it's pretty much the same. Additions to the new video player software include video stream selection (the different angles button on your old DVD remote) and the ability to better hand off video playback tasks to add-ons. This isn't something you're going to notice day-to-day, but the real achievement is that the previous system was completely replaced with the new one without collapsing Kodi in the process: it being pretty much the same means they did their job. It's a good thing.

One thing we did notice was that the video output from the HDMI worked a little better with the HDMI switcher in our setup: in earlier versions, audio from a Pi using Kodi would sometimes hiccup as if the audio stream itself just wasn't making it to the TV. We didn't come across this issue with Kodi 17. There's still a little trouble playing video encoded with 10-bit colour, though, with no improvements over the previous release. That's more of an issue with a lack of hardware decoding for 10-bit on the Pi than with Kodi.

kodi.tv

Free



Left Add-ons can now also be found in a dedicated add-on category, though some will still show up in the relevant media categories

We also noticed that, while still supposedly supported, AirPlay was working worse than ever. Before, an Apple device could at least see the Kodi box although we've never actually seen AirPlay working between the two. Now, it simply doesn't show up. It would be good if the Kodi guys could get that working consistently, but it's a minor issue.

It runs beautifully on the Raspberry Pi, though, with smooth transitions between menus and the same basic structure as other skins.

> It runs beautifully on the Raspberry Pi, with smooth transitions between menus

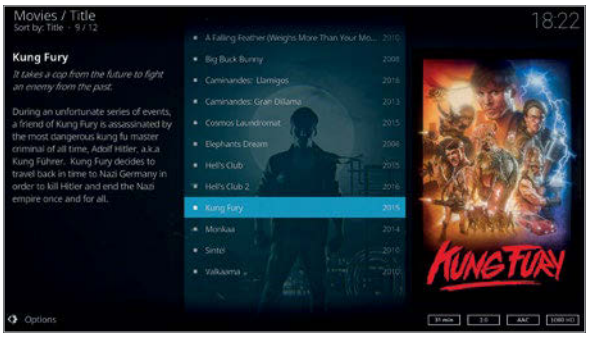
New look

That brings us to the new interface, Estuary. It's almost completely different to the old Confluence, with a greater focus on using the scraped metadata from your content and optimising the space to display said content. Inspiration has definitely been taken from modern smartphone UI and UX design, which makes sense as the way we use technology has changed over the years. You can still have your basic file selection if you prefer, though, and you actually get more control over which categories are listed on the home interface. That being said, you can't change the order in which they're listed and, for a device you're probably mostly using to play back video, having the pure video option last in the list is not very useful.

The settings menu is generally more useful and easier to navigate as well, although it does mean some extra add-on settings now live away from the settings menu in an all-encompassing add-ons field.

Future-proof

This new version of Kodi seems like a glimpse of what's to come. With a more robust back end and a new front end to go with it, we're probably going to start seeing a few more noticeable changes in the next version of Kodi. The new interface is great and a welcome change over the old Confluence one (although you can still install it if you prefer). Importantly, it still runs really well on a Raspberry Pi. We did most of our testing on a Raspberry Pi 2, which is more than likely the backbone for any home theatre setup, and it worked as well as ever, so there's no worries there.



Above The new interface is arranged much better than before, and even has symbols to signify if you have watched or are part-way through a video

It's definitely a worthwhile upgrade over previous versions, so if you're a LibreELEC user you should be on the lookout for version 8 when that comes out. Other HTPC OSes may be a little slower to integrate the new stuff, so you shouldn't have to worry about it just yet.

Last word

A good upgrade to an already great piece of software. It still works just fine on the Raspberry Pi, so there's no need to overhaul your home theatre for it either.

★★★★½

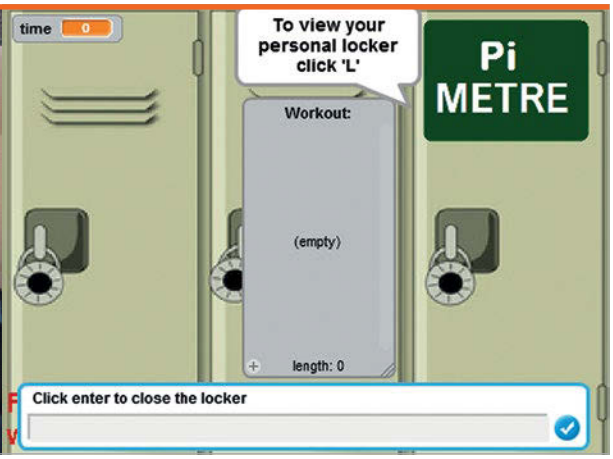
bit.ly/pimetre

FREE

Maker Says

"Want to lose the Xmas dinner? Join Pi Metre"

Pi Metre



PI METRE

A Scratch-powered fitness app for the Raspberry Pi. How does it help you get fit?

Related

MY FITNESS PAL

While MyFitnessPal is great, it doesn't run on the Raspberry Pi. The app is available for iOS and Android.

FREE

myfitnesspal.com

The Pi Metre is a simple fitness app created for Raspberry Pi in Scratch. It was designed completely on Scratch 2.0, so it's only recently been able to work widely on Raspberry Pis thanks to the addition of Flash to Chromium. Luke Castle, the creator of Pi Metre, suggests also making a portable Pi setup to take it with you to the gym or while running, but it would work just as well at home.

Pi Metre works by asking you to first start a workout. You do this by pressing **E** and entering the number of seconds you plan to work out; this does slow down the process a bit, as you need to transfer the minutes you plan to work out into seconds. The app will then count down the seconds and play an appropriate workout mix for you to exercise to. It lasts just over five minutes before looping again, so for longer workouts it may get a bit repetitive. Once the countdown is done, it triggers an alarm to let you know your workout is over.

> "You can create a portable Pi setup to take it with you to the gym or while running"

Workout times are then kept in your locker, which can be accessed with **L**. Unfortunately, there's no way to differentiate between different types of workouts, and it won't calculate a calorie count for cardio. You get an account for the app as well, with the idea that you can save your workouts to refer back to them; you'll need to make sure to log off, though, to save the workouts properly.

It's a very simplistic app and could do with a fair few extra components. Also, as it runs on Scratch, it will use up a lot of resources. It's quite fun, though, and a nice little project to make so you can create your own portable workout countdown and tracker.

Last word

A very simple workout app which is much better as a full Pi project than just the software on its own. You're probably better off with a smartphone app, though.

★★☆☆☆

magpi.cc/2kXdZsE

£12 / $15

Maker Says

"Turn your Pi into a Lilliputian ghetto blaster"

Pimoroni

SPEAKER PHAT

A tiny speaker board for adding sound to projects

Pimoroni's latest pHAT may not sound all that phat, but it does offer a neat way to quickly add audio to your projects. This Pi Zero-sized board manages to cram in a mini speaker, I²S DAC, and mono amplifier. We particularly appreciate the 1980s boombox-style artwork on the top of the board, complete with a ten-LED bar graph.

Since the 8Ω speaker has just 2W of power, you won't exactly be rocking the neighbourhood. It sounds more like a tinny transistor radio than a ghetto blaster, offering little in the way of bass (and obviously no stereo), so it's certainly no rival for high-quality audio add-ons from the likes of IQaudIO and JustBoom. But then that's not what it's intended for; instead, it should prove suitable for projects that require audio output for notification sounds or speech. Indeed, it would be ideal for the digital whoopee cushion in this issue, precluding the need to wire up a separate powered speaker. Other use cases include a retro gaming handheld or – by pairing it with a mini microphone – a DIY voice assistant based on Amazon Alex or the like.

> "It is nevertheless a great way of adding audio to projects"

The Speaker pHAT comes in kit form, requiring about 30 minutes to assemble. First, you need to screw the speaker to the rear of the board, using tiny spacer nuts, which is a little fiddly. Then it's time to get the soldering iron out. Soldering the speaker's two terminals to the board contacts with the supplied 24AWG wire is a bit awkward and you need to use tweezers to hold the bare wire as it gets hot! It's then simply a matter of soldering on the 40-pin female header and the pHAT is ready to use.

Installation of the software is achieved with a single terminal command. It should then work as the default audio output, although you may need to reboot the Pi first. We tested it out with aplay and mplayer from the command line, as well as Sonic Pi and YouTube in the Chromium web browser. There's also a library for manually controlling the LEDs if you want.

Last word

Not one for audiophiles, the Speaker pHAT offers rudimentary sound quality with little power. It is nevertheless a great way of adding audio to projects, with its tiny Zero-size footprint and no need for a separate power supply. It looks cute too.

★★★★☆

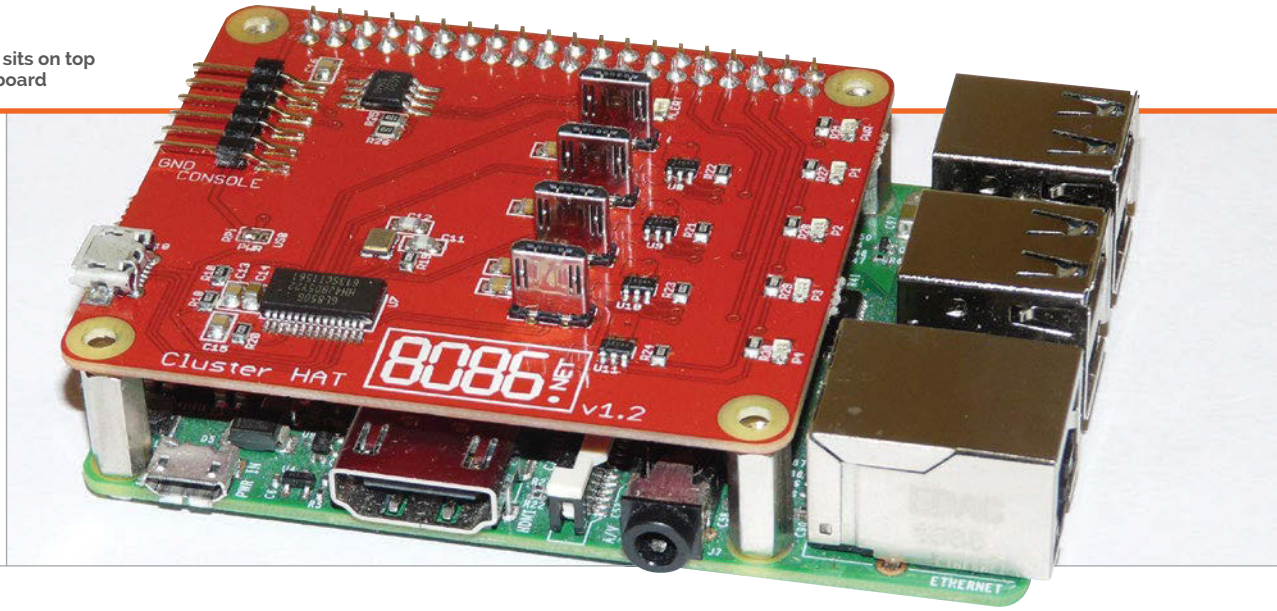
Related

MINI PORTABLE SPEAKER

Powered or charged via USB and connecting to the audio out port of a standard Pi (but not Zero), this mini speaker packs a punch.

£12 / $15

magpi.cc/2diQyHq



Right The Cluster HAT sits on top of a Raspberry Pi 2/3 board

Maker Says

"Ideal tool for teaching, testing or simulating small-scale clusters"

pimoroni

CLUSTERHAT

Make a cluster computer with a Raspberry Pi and four Pi Zeros

Building a computer cluster is one of the most impressive Raspberry Pi projects. And being able to program cluster computers is one of the most highly valued skills in the world of big data.

A cluster is a set of computers networked together and used as a single system. Each computer (or 'node') is scheduled to work on the same task, controlled by a master computer.

Knowing how to manage, and use, clustered computers and supercomputers is a pretty valuable skill. It's also one of the areas at the absolute forefront of computing technology.

But it has always been difficult to get started. Traditionally, cluster computing has been an expensive hobby. You need lots of computers (preferably identical) and a network controller and the skill to network them all together. This difficulty is before you even start learning how to manage several computers working in tandem.

On top of all that, it can be expensive to run multiple full-sized computers. Energy costs quickly ramp up when you're running a cluster.

Hat on it

The Cluster HAT removes the expense and effort required to create a cluster. This lower barrier to entry makes it a lot easier to get on with the more important act of learning cluster computing technologies.

The HAT sits on top of a Raspberry Pi 2/3 and you slot four Pi Zero computers on to the HAT (via the USB connection).

A USB cable is run from the Raspberry Pi to the HAT to provide power to the four Pi Zero boards. The latter are controlled directly via the GPIO pins.

You need a microSD card for each Pi Zero, and one for the controlling Raspberry Pi. So that's five SD cards in total.

We costed up a Raspberry Pi, Cluster HAT, four Pi Zero boards, and four official microSD cards – it comes to £102. That's by far the cheapest cluster computer you will find on the market. For those looking to get into cluster computing, this is a game-changer.

Getting started

Setting up the Cluster HAT hardware was an absolute breeze. Screw the four pins on to a Raspberry Pi 3 board and insert the supplied USB cable, then attach the HAT (wiring the cable between the board and HAT). Attach the USB power to the HAT and then slot on the four Pi Zero boards. The whole setup took less than five minutes.

Software is well handled too. A dedicated site (**magpi.cc/2jHMeQq**) provides image files for the controller (the Pi 2/3) and one image file for each Pi Zero board.

The Cluster HAT version of Raspbian boots into the command line, but offers a script for controlling the cluster. For example:

Related

PICO 5 STARTER KIT

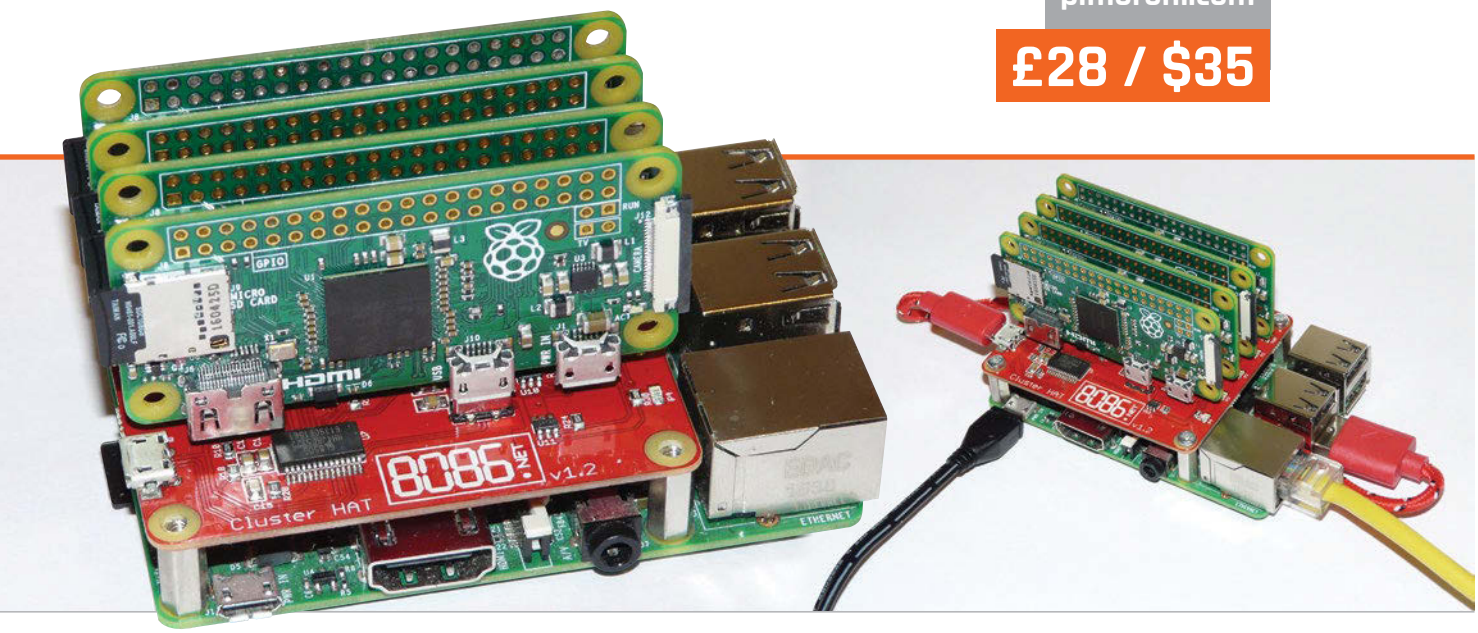
PicoCluster has a range of cluster kits for a range of boards (including the Raspberry Pi). They use Ethernet networking and Raspberry Pi 2/3 boards, but are more expensive.

£189 / $239

magpi.cc/2kAcJLG

pimoroni.com

£28 / $35




```
clusterhat on all
```

This command-line code switches on all four Pi Zero boards. You can also control them separately, using **p1-4**. Such as:

```
clusterhat p2 off
clusterhat p2 on
```

It takes a few moments for each Pi Zero to power up and load its software, at which point it'll appear in **ifconfig**.

Access each node using SSH (enabled by default):

```
ssh pi@p1.local
```

Here you can use **ifconfig** to find the IP address Cluster HAT has assigned to the Pi Zero (under USB Ethernet).

Pi computing

One thing this isn't, though, is a supercomputer. There's a good set of benchmarks by Nick Smith on Climbers.net (**magpi.cc/2jKlq23**).

Nick used **iperf** to test the network-over-USB connection and found it comparable to Ethernet.

The slower speed of the Pi Zero boards compared to Raspberry Pi 2/3 boards is much more pronounced.

Nick benchmarked the system using HPC Challenge Benchmark (**magpi.cc/2jKpH5C**). He found the cluster of four Pi Zero boards ran at roughly half the speed of a single Raspberry Pi 3 board.

The slower speed makes sense when you think about it. Nick explains: "This is partly down to the faster individual cores in the Pi 3, but also the node-node communication is obviously a great deal faster within the Pi 3 SoC compared to communicating between separate Zeros with Ethernet-over-USB."

So you're not building a faster Raspberry Pi. However, we don't think this matters. The aim for most people will be to build a low-cost and relatively easy-to-assemble cluster for educational purposes.

Once created, you can start to experiment with the concepts behind cluster computing. The Cluster HAT is ideal for teaching, learning or developing cluster computing ideas.

We suspect it will also appeal to developers who have access to larger cluster computing systems, but not all the time. You can test and run code on the Cluster HAT, then migrate it across to a larger system.

Distributed computing

Cluster computing is not a game for beginners. Realistically speaking, prospective learners will need a working knowledge of Python, Java, SQL (and relational database techniques). You'll also need a good understanding of the Linux command line.

Fortunately, starters are not alone. There's a vibrant Cluster HAT Google Groups board (**magpi.cc/2jHYqk6**). There are lots of resources and projects being shared by the community.

One book that is recommended reading is *Raspberry Pi Super Cluster* (£16.78) by Andrew K Dennis (**magpi.cc/2jHZyV8**).

Whatever resources you choose, we think the Cluster HAT is a great resource for starting distributed computing. It's small, cheap, and easy to set up.

You have to leap a lot of hurdles to learn cluster computing. But Cluster HAT removes the expensive and difficult hardware build. It's then up to you to learn the sophisticated coding techniques.

Last word

A compact, neat and cost-effective solution to building a cluster computer. The end-product might lack raw power, but it's perfect for teaching and learning distributed computing concepts.

★★★★★

Above Left The USB sockets on the Cluster HAT provide connection for four Pi Zero boards

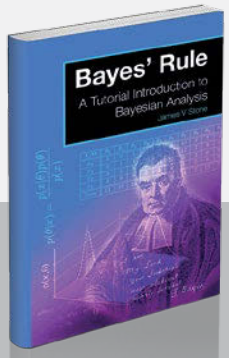
Above Right The Raspberry Pi 3 board is the only device using Ethernet. Network-over-USB is used to hook up the Pi Zero boards

TECHNICAL BESTSELLERS

SEBTEL PRESS

Dr James Stone's Tutorial books for Sebtel make technical, mathematical subjects easily understandable.

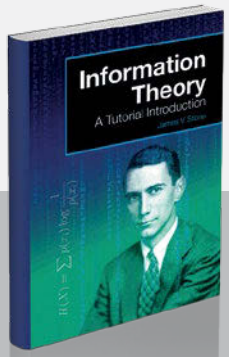
BAYES' RULE: A TUTORIAL INTRODUCTION TO BAYESIAN ANALYSIS



Author: Dr James Stone
Publisher: Sebtel Press
Price: £14.95
ISBN: 978-0956372840
magpi.cc/2lFLqx5

The revival in interest in Bayes' theorem – studying probability based on information we have – has found a use in many fields. This is possibly the best general introduction to the field. Recommended.

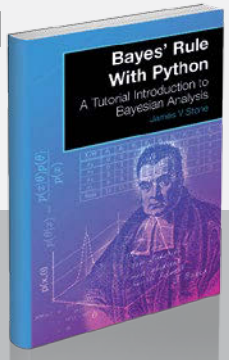
INFORMATION THEORY: A TUTORIAL INTRODUCTION



Author: Dr James Stone
Publisher: Sebtel Press
Price: £16.95
ISBN: 978-0956372857
magpi.cc/2lFOwRp

An outstanding introduction that can pull in the general reader with dry humour, but has the textbook rigour needed by programmers looking for a fuller understanding of this essential subject.

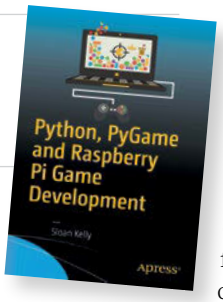
BAYES' RULE WITH PYTHON



Author: Dr James Stone
Publisher: Sebtel Press
Price: £19.95
ISBN: 978-0993367939
magpi.cc/2lFLqx5

The Bayes book, with inline Python code examples of the probability problems. Also available in R and Matlab flavours. Very handy for helping to work through tougher examples in Jupyter Notebook.

PYTHON, PYGAME AND RASPBERRY PI GAME DEVELOPMENT



Author: Sloan Kelly
Publisher: Apress
Price: £14.99
ISBN: 978-1484225165
magpi.cc/2lFOdGy

Writing games remains one of the best ways to motivate learners through some of the challenges of learning to code. The Pygame library provides enough abstraction to let the reader produce working games relatively easily, without hiding away anything of real educational value. Marketed at Pi users, Kelly's fine tutorial can be used on any computer where Pygame is installable.

Hello World is preceded by a useful introduction to data types and programming styles – who knew pencils were object-oriented? – then followed by programming basics, all with reference to parts of typical games. The writing style and short chapters give a pleasing forward momentum, and we're soon at our first game.

After due recognition to design and testing, there's Bricks, a classic Breakout-style game; always a good teaching example, with movement and collision. Building on this example, object-oriented programming is explored fully; there's a lot here, but some of it gets reinforced in the next game, Snake. MVC is introduced for Robots. Final chapters cover sound and finite-state machines, wrapping them up with a game of Invaders. Not always easy going, but this well-paced introduction to programming will rapidly carry newbies a surprisingly long way.

Score ★★★★☆

THE BIG BOOK OF MAKERSPACE PROJECTS

Authors: Colleen Graves, Aaron Graves
Publisher: McGraw Hill
Price: £15.99
ISBN: 978-1259644252
magpi.cc/2lFLiNM

Learning programming has well-established routes, such as Hello World, loops, and variables, but what's the beginner maker to do? The Graves' excellent collection of projects will build up your skills, then challenge you to go further. Hello World here is the Brushbot Warrior, repurposing everyday equipment, then proceeds via Scribble Bots to a host of other moving projects.

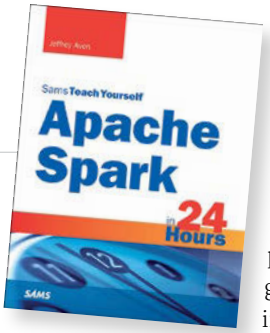
After circuits, coding, and smartphone projects (DIY projector), we're back to non-tech skills, with some great musical instrument builds. After sewing circuits, more electronics; the supply listings are localised to US suppliers, but it shouldn't take too much effort to track things down wherever you are, locally as well as through the internet. Projects are followed by challenges, suggesting you step beyond following instructions and make projects your own.

Every maker needs some of these projects in her armoury, ready to build with the next person who asks "what's it all about, then?". If you're new to making, there are plenty of cheap and easy projects to build skills and confidence. Many of these will be great to run on makerspace open days. And if you're an old hand, the challenges and harder projects are well worth trying out.

Score ★★★★☆

APACHE SPARK IN 24 HOURS

Author: Jeffrey Aven
Publisher: SAMS
Price: £27.99
ISBN: 978-0672338519
magpi.cc/2lFUBot



Apache Spark is an essential tool for tackling big data, providing an API for data processing in Scala, Java, R, or Python. Aven's book focuses on Python and, beyond a basic comfort in that language, assumes no particular knowledge on the part of the reader. It introduces everything from Scala to Hadoop before tackling a range of Spark topics with Python examples.

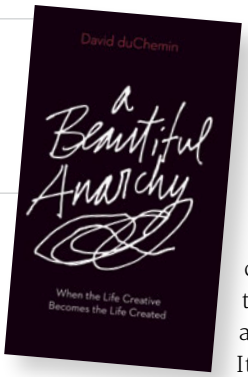
The tools of the trade for big data and Spark are introduced one by one, from Hadoop and MapReduce, through Resilient Distributed Datasets (RDD) and cloud computing, to machine learning and message queues. RDDs are in-memory data structures which are part of Spark's speed and efficiency; the Spark Core API is the other key to Spark's success, respecting a strongly functional style of programming.

R, SparkR with RStudio, and Scala get chapters, and it's great to see functional programming in Python getting treated as well as in Scala; however, that chapter also squeezes in IPython and Jupyter Notebooks, showing the limitation of the 24 hour-style chapters. Final chapters on administering, extending, monitoring, securing, and improving the performance of Spark give the reader everything she needs to go further – a solid foundation for Spark with Python.

Score ★★★★☆

A BEAUTIFUL ANARCHY

Author: David DuChemin
Publisher: Rocky Nook
Price: £14.99
ISBN: 978-1681982342
magpi.cc/2lFTBt9



A very personal manifesto, written for anyone with a calling to make, whether it's a painting or a great business. "We're all creative," says the author, "but we've allowed art to co-opt the word." Many take default choices rather than the difficult path of trying to write that novel they know they have in them, or set up that business. This is a call "to step out of the comfort zone, and face the fear."

Praising bad ideas, constraints, and the art of exclusion, DuChemin returns to the same theme from a different angle: you have a creative urge inside you, so stop making excuses, and start facing up to the difficult journey to writing that novel, starting that business, or painting that picture: "Living a great story is much harder than watching a great story."

This is a book about choices; those we make or those we abdicate all have an effect on our creations. It gets the reader to ask themselves questions about what they want in life. What dreams do you still hold onto? This call to colour outside the lines, and DuChemin's questioning, have a way of getting under your skin and making you ask those difficult questions.

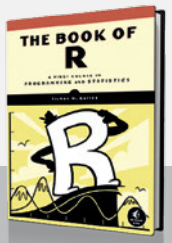
Score ★★★★☆

ESSENTIAL READING: R FOR STATS

While Python nips at R's heels, R remains great for statistics and many general data tasks.

The Book of R

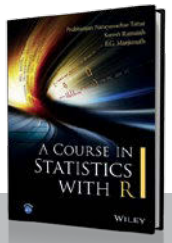
Author: Tilman M Davies
Publisher: No Starch
Price: £39.99
ISBN: 978-1593276515
magpi.cc/2lG0Ykf



Combines a programming tutorial with an introduction to statistical methods, succeeding well at both tasks. Strongly recommended for newbies.

A Course in Statistics with R

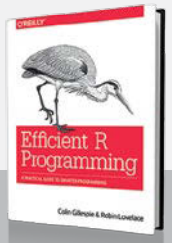
Authors: Prabhanjan N Tattar, Suresh Ramaiah, B G Manjunath
Publisher: Wiley
Price: £70.00
ISBN: 978-1119152729
magpi.cc/2lFV5Uj



Despite occasionally idiosyncratic English, this is a wonderful deep dive into statistics strongly integrated with R code.

Efficient R Programming

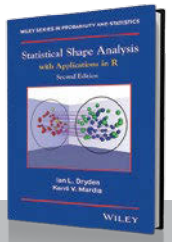
Authors: Colin Gillespie, Robin Lovelace
Publisher: O'Reilly
Price: £28.50
ISBN: 978-1491950784
magpi.cc/2lFOUzE



Essential next step in learning R – efficiency in programmer (your) time, and eliminating bottlenecks through deeper language understanding.

Statistical Shape Analysis: With Applications in R

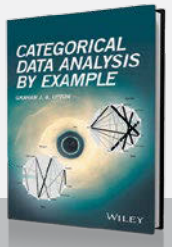
Authors: Ian L Dryden, Kanti V Mardia
Publisher: Wiley
Price: £65.00
ISBN: 978-0470699621
magpi.cc/2lFK1GK



Statistical methods applied to shape analysis. Great for biologists, but strong mathematical treatment and accompanying code expands possible applications.

Categorical Data Analysis by Example

Author: Graham J G Upton
Publisher: Wiley
Price: £80.50
ISBN: 978-1119307860
magpi.cc/2lG6930



Concise introduction to dealing with categorical data (with supporting R code) which will help the general data scientist.

FOSDEM

EUROPE'S INCREDIBLY BIG FREE SOFTWARE DEVELOPERS' MEETUP

In medieval times, 10,000 was the population of a major city. For one weekend each February, it's the number of free software developers attending FOSDEM in Brussels

Ten thousand attendees is a guesstimate, as FOSDEM is an entirely free event with no registration. It's certainly very big! Now in its 18th year, FOSDEM has been referred to as 'a week-long event happening over a weekend'. Indeed, it feels very concentrated, with dozens of talks happening at once (610 speakers in total), almost all rooms full to capacity, and busy project stands spread over several buildings.

The conference runs on goodwill, volunteer energy, and a modest amount of discreet sponsorship (plus coffee and Belgian beer). While all talks are in English, and many Americans fly in to attend or speak, the corridors buzz with geeky chat in Flemish and French, and Scandinavian and Slavic languages abound. This is *Europe's* developer meetup; it exists because it's important for free and open source software developers to be able to meet in real life.

With 55 tracks over 30 rooms of the Université Libre de Bruxelles, you'll find not just your favourite projects, but many new discoveries. This year's event included languages from GNU Guile to Python, technologies from containers to microkernels, and hardware projects covering everything from the embedded space to chip designers. Add in documentation, design, and community management, and FOSDEM has the entire free software and open hardware ecosystem present.

Introductory music

There's no room on these pages to even list the 600+ talks, so here's a tiny flavour of all that was on offer. Let's start with the arts. Music in particular is often well-represented at techie events, and one outstanding free software project is MuseScore, which makes it easy to produce great-looking sheet music that can also be played back through the project's soft synthesizer.

The project featured in the design track, as its usability research was profiled. It was also featured in a talk on OpenScore, which is crowdsourcing MuseScore digitisation of out-of-copyright sheet music to build a vast library of semantic score data, open for analysis with tools like MIT's music21, as well as freely accessible sheet music. Partners for OpenScore include the IMSLP public domain sheet music library: **imslp.org**.

Good as it was to see some old-school analogue synths in the slides of Python developer Yann Gravrand, it was even better to hear a live demo of them being controlled via Python by his old MIDI drum pads, and the mido and portmidi libraries.

More exotic languages were used by Belgian digital artist François Zajéga in the Open Game Development devroom. He demonstrated sounds from the Pure Data language, with Ogre3D visuals on the Polymorph video game platform, suitably impressive visuals and music, dancers, and artists. His presentation is available here: **magpi.cc/2louvIf**.

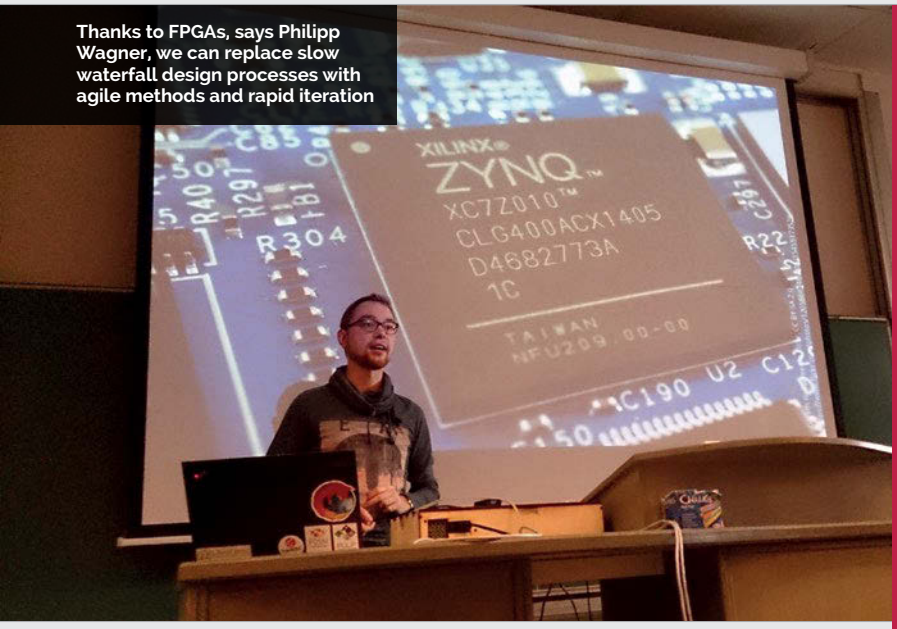
There were more visual arts from swarms of lightweight (27 grams – less than an ounce!) Crazyflie 2.0 drones, painting, climbing, and commanded by a MIDI glove. Drones were also flying over the attendees in the GoLang rooms, where systems programming included IoT.

OPENING THE LEDGER

Free software developers don't just pick the fun stuff – Software Freedom Conservancy was discussing its work on new ledger-based accounting software for not-for-profit organisations, something the third sector has been demanding for many years.

And on a global scale, Evan Schwartz was talking about tackling the mess of incompatible digital payment systems with W3C's InterLedger project, an inter-networking protocol for money, with many parallels to the way the internet itself works.

The slides and white paper are available here: **magpi.cc/2lovKHy**



Thanks to FPGAs, says Philipp Wagner, we can replace slow waterfall design processes with agile methods and rapid iteration

FREE CHIPS (NOT FRITES)

The dependency of free software on far from open hardware has been a nagging worry since the early days of the movement, and projects like LibreBoot are perennial FOSDEM visitors. This year's event saw two highlights for those looking to run on entirely free and open hardware.

Georg Greve, CEO of Kolab and former FSFE president, impressed the audience with the potential of OpenPOWER chips for high-performance computing and the data centre, while the less powerful but very adaptable OpenRISC V offered many custom and off-the-shelf possibilities.

In addition to these, Olimex had brought its TERES-I DIY laptop from Bulgaria, complementing modularity and free software with free tooling through the design process, and open-source PCB CAD files on GitHub.

Pi's subtle presence

The Raspberry Pi is very well established to the point that it's not much talked about. However, it was quietly working on many displays, such as Automotive Grade Linux (running on Yocto and OpenEmbedded), in ports of much of the software from the Kolab images to experimental builds of Haiku and Illumos, and in NextCloud's Pi 3-based Dropbox replacement.

Ben Nuttall was there to fly the flag for the Raspberry Pi Foundation, and talk about HATs. If you're interested in building your own HAT, with entirely free and open-source tools including the board design, watch the video of Leon Anavi's talk. And look out for a feature on Anavi's HAT in these pages soon.

Python got its own full day of lectures: everything from how data structures are implemented in CPython, to Python in the cloud. Roberto Colistete Jr talked about 'Scientific MicroPython for Microcontrollers and IoT'. BoF sessions were held outside the lecture room, and posters revealed how much internet giants like Facebook use Python. It's a sort of mini PyCon, with bonus sidetracks.

Etiene Dalcol spoke about making wearables with NodeMCU, a Lua-based language. Interesting 'niche' languages were well-represented, although the regular visit from Belgium's Ada group is sadly a thing of the past. Guile, the Scheme dialect used in many GNU projects and particularly in GUIX, featured many interesting sessions. Of these, *MagPi* readers may be interested in 'Natural Script Writing', featuring a Guile dialect with whitespace, which "makes the transition from Python to Lisp seem a lot less scary."

Sharing knowledge

Documentation is a concern for anyone trying to get things done with a piece of software. Rich Bowen had some great advice for those responsible, stating that documentation – including IRC channels – sets the tone for your project. How welcoming are you to new users? If you find new users not sticking with your project, take a listen to his presentation.

At the top end of the document-handling scale, XWiki was doing a sterling job of presenting itself as a useful alternative to Microsoft Sharepoint. The humble TikiWiki was also present as part of the WikiSuite tools, offering secure, self-hosted software for 'data-driven' organisations.

Knowledge was on offer everywhere: a chance hallway meeting with Nikita Voloboev offered the chance to try large mind maps of scientific knowledge: **magpi.cc/2loQNcP**. Sharing must be balanced with privacy, and pp (pretty Easy privacy) was pushing easy cryptography for the masses. The crowded Decentralised Internet devroom also offered a roadmap for the next few years' defence of privacy.

Daniel Stenberg, the creator of the ubiquitous tool **`curl`**, gave a talk named 'You know what's cool? Running on billions of devices', about the unexpected demands of maintaining code that everybody's using. There were talks on legal issues and community development, lightning talks, considered keynotes, file formats in open media, and data formats in many languages. Some of the most interesting talks were in the Software Defined Radio devroom.

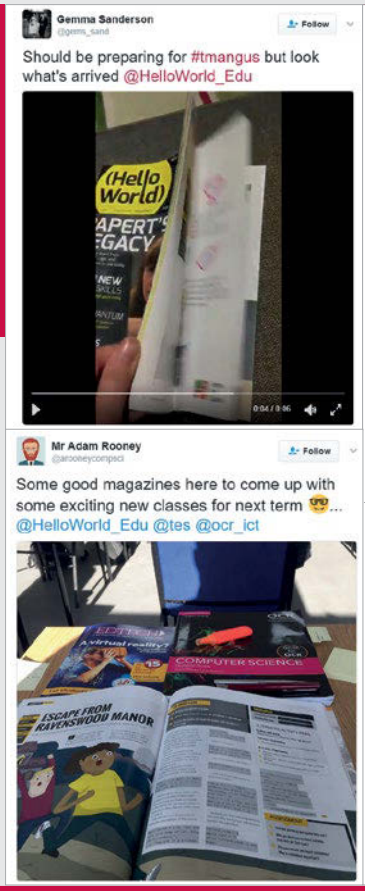
FOSDEM has evolved over the years, gradually embracing some elements of the upstart maker community, as well as embedded development (particularly Automotive Grade Linux), and even chip design. By far, the best of FOSDEM remains the corridor conversations, which are a great opportunity to meet programmers and other free software advocates of all backgrounds. But if you missed it, you don't have to wait for next year – take a look at the magnificent archive of technical talks at **fosdem.org/2017**.

THE MONTH IN RASPBERRY PI

Everything else that happened this month in the world of Raspberry Pi

HELLO, WORLD!

We launched a sister magazine last month called *Hello World*, a free computing magazine aimed specifically at educators. The reactions to it have been fantastic, with attendees of the Bett Show and other educators around the UK (and indeed the world) excited to get their hands on a copy. You can find out more info on how to get one yourself at **helloworld.cc**, and help kids and teens learn more about computing everywhere.


Gemma Sanderson
Should be preparing for #tmangus but look what's arrived @HelloWorld_Edu
Richard Smith
Used the great helloworld magazine with a small group of students. Wonderful ideas. Free for educators @mberry @HelloWorld_Edu @CompAtSch
Hello World Magazine
Mr Adam Rooney
Some good magazines here to come up with some exciting new classes for next term ...
@HelloWorld_Edu @tes @ocr_ict
Cloghans Hill NS
Thank you so much @magsamond we r thrilled with our special delivery of @HelloWorld_Edu it's amazing

HIDDEN FIGURES

More secret history from behind the scenes of the Apollo missions

You may remember a few issues ago that we did a piece on using the Raspberry Pi to run the Apollo Guidance Computer code. This was after the code for it had been uploaded to GitHub in July 2016, a wonderful mixing of old code and new tech. One of the things we talked about extensively was Margaret Hamilton, one of the head coders behind the project. It's because of her that the concept of software engineering exists today, but she wasn't the only person breaking boundaries at NASA in those days, as a reader on Twitter pointed out to us. *Hidden Figures* is about these people: the women of colour such as Katherine Johnson working behind the scenes as mathematicians, performing orbital calculations among many other things for NASA. It's been nominated for three Academy Awards, including Best Picture, and shows the struggles these women went through, not just because of their gender but also the colour of their skin. It's an excellent watch and we recommend it.

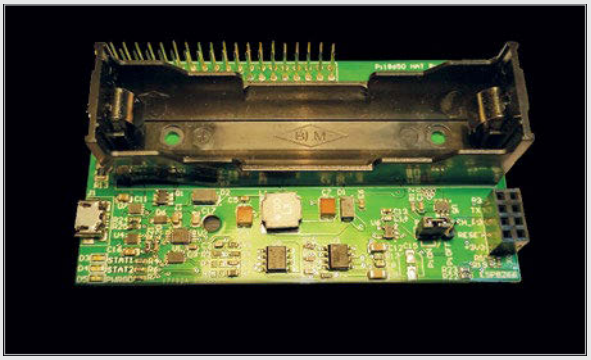
CROWDFUND THIS!

The best crowdfunding hits this month for you to check out...

MEARM PI

kck.st/2kIlKTB

If you cast your minds back a little while, you may remember that we reviewed the excellent MeArm robot arm – it's pretty popular in the Pi community, even though you need to do some rejigging to get it working with it. It's no surprise, then, that the team behind it are making a Pi-specific version, and have asked the community to help fund it, which they did in less than 12 hours. They're currently at 300%, so it's a great time to jump on and get your own kit.



PI18650 HAT BATTERY HAT

kck.st/2l6bwts

While not boasting a very descriptive name, the Pi18650 HAT allows you to plug in 18650 rechargeable Li-ion batteries to power the Raspberry Pi. The HAT fits on older 26-pin Raspberry Pis as well, and offers a new way of enabling portability for the Pi. It's smashed its funding target, so get on board now to get one to boost your portable projects.

BEST OF THE REST

Here are some other great things we saw this month

magpi.cc/2kQAOfw

GBAPI

We've seen lots of Game Boy conversions and even custom builds (like our modified PiGRRL in the cover feature), but this is possibly the first GBA conversion we've seen using RetroPie. Although it's missing two buttons to be able to go full-on SNES, you could probably mod one a bit further and add buttons, and the backlight the original lacked. We love it!

magpi.cc/2kQJgjn

RUNMYROBOT

We like Let's Robot and apparently other people in the community like the idea as well – there's now an entire website dedicated to letting people give random strangers on the internet access to robots in their home. You can add your own if you want, albeit at your own peril.

magpi.cc/2kQp5NT

GUEST WIFI

A makerspace in Santa Barbara, CA is automating guest WiFi codes using a Raspberry Pi, a thermal printer, and a big old button. Press it for a 24-hour code you can use in the makerspace. It's simple and easy. We should get one for when people come over and want the WiFi code.

COMMUNITY PROFILE

DAVID PRIDE

A Raspberry Pi maker and student, who plans to add as many letters as he can after his name

David Pride

Category: Maker and student

Day job: Currently getting paid to study what he loves

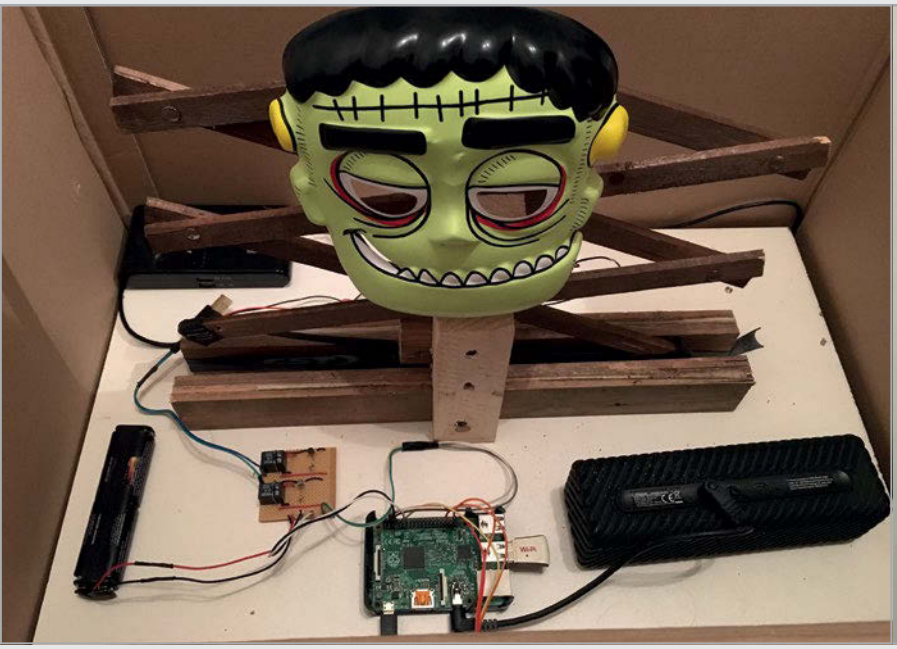
Website: piandchips.co.uk
twitter.com/davejavupride

David's experiences in computer education came slightly later in life. He admits to not being a grade-A student: he left school with few qualifications, unable to pursue further education at university. There was, however, a teacher who instilled in him a passion for computers and coding which would stick with him indefinitely.

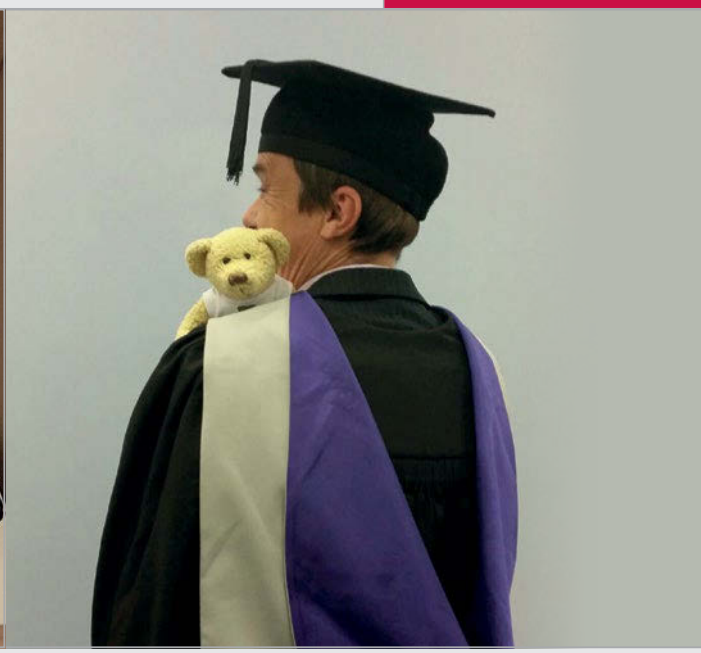
Twenty years later, back in 2012, David heard of the Raspberry Pi – a soon-to-be released "new little marvel" that he instantly fell for, head-first. Despite a lack of knowledge in Linux and Python, he experimented and had fun. He found a Raspberry Jam and, with it, Pi enthusiasts like Mike Horne and Peter Onions. The projects on display at the Jam were enough to push David further into the Raspberry Pi rabbit hole and, after working his way through several Python books, he began to take steps into the world of formal higher education.

With a MOOC qualification from Rice University under his

Below David joined us at the St James's Palace Community celebration, mingling with the likes of the Duke of York, plus organisers of Jams and clubs, such as Grace and Femi



Above Self-playing xylophones, Wiimote air drums, Lego sorters, Pi Wars robots, and more. David is continually hacking toys, giving them new Pi-powered life



Above David's determination to access and complete further education in computing has earned him a three-year PhD studentship. Not bad for a "lousy student"

belt, he continued to improve upon his self-taught knowledge and was fortunate enough to be accepted to study for a master's degree in Computer Science at the University of Hertfordshire. With a distinction for his final dissertation, David completed the course with an overall distinction for his MSc and was recently awarded a fully funded PhD studentship with the Knowledge Media Institute.

> "Despite a lack of knowledge in Linux and Python, he experimented and had fun"

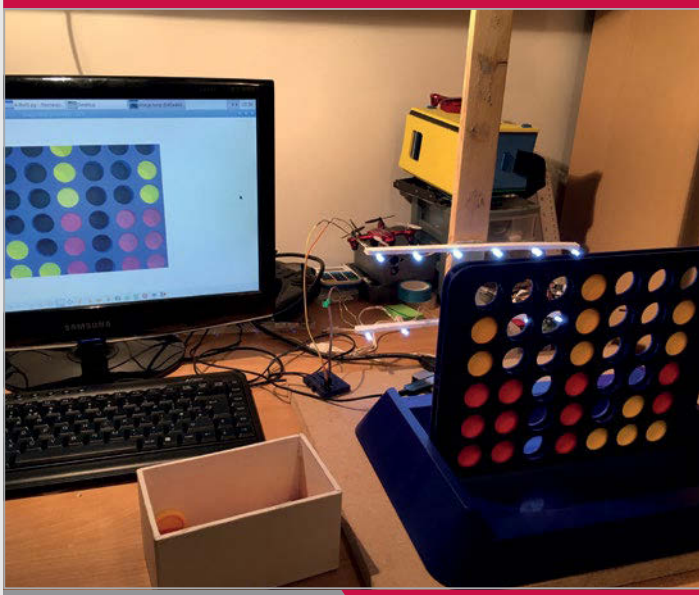
The portfolio of projects that helped him to achieve his many educational successes has provided regular retweet material for the Raspberry Pi Twitter account, and various blog entries have been penned to highlight his fun, imaginative works.

His builds have travelled to a range of Jams and made their way to the Raspberry Pi and Code Club stands at Bett, as well as to the birthday celebrations for the tiny microcomputer.

His website, the pun-tastic Pi and Chips, is home to the majority of his work; David also links to YouTube videos and walk-throughs of his projects, and relates his experiences at various events.

If you've followed any of the action across the Raspberry Pi social media channels – or indeed, read any previous issues of *The MagPi* magazine – you'll no doubt have seen a couple of David's projects. The 4-Bot, a robotic second player for the family game Connect Four, allows people to go head to head with a Pi-powered robotic arm. And believe us when we say that it's not as easy to beat as you'd hope. Then there's his more recent air drum kit, which uses an old toy found at a car boot sale, plus a Wiimote, to make a functional air drum displaying David's toy-hacking abilities… and his complete lack of rhythm. He does fare much better on his home-made laser harp, though!

HIGHLIGHT

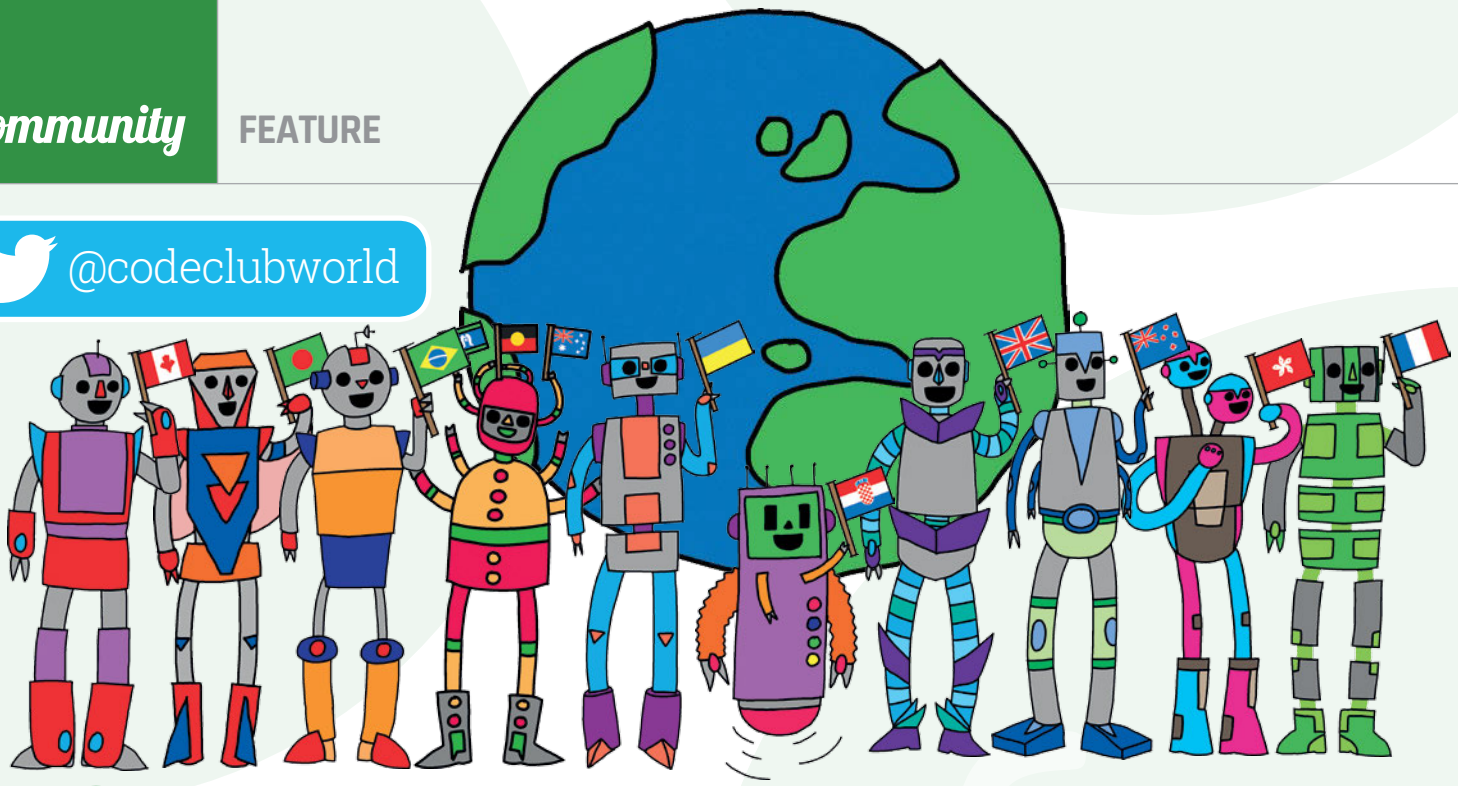


magpi.cc/1XrC3zU

4-BOT

It seems only fair that we talk about the 4-Bot. Many readers will have seen this wonderful build online, where it has made its way across various blogs and websites, and even made an appearance alongside David in a recent Bloomberg interview. Using a Python imaging library, the 4-Bot splits the game grid into 42 squares, and recognises them as being red, yellow, or empty by reading the RGB value of the space. Using the minimax algorithm, 4-Bot is able to play each move within 25 seconds. Considering the trillions of possible game positions, David made a compromise and, if you're lucky, you may just be able to beat it.

@codeclubworld



CODE CLUB INTERNATIONAL

Code Club helps get kids coding around the world. Here's how...

If you haven't yet heard about it, Code Club is a global network of volunteer-led extracurricular coding clubs for children.

Code Club first started in the UK, back in 2012, to help children develop coding skills. Since then, this idea has spread around the world and there are now over 9,000 Code Clubs internationally.

In 2016, Code Club officially launched its international programme (Code Club International, **magpi.cc/2lfaGNm**). Code Club partners with local NGOs (not-for-profit organisations). These share the mission to have a Code Club in every community in the world, and support the country's local Code Club community.

Currently there are Code Club partners in ten countries. Code Club Australia, the biggest community outside the UK, has over 50,000 children attending Code Clubs (**magpi.cc/2lgjyiE**). In fact, in 2016 they set a world record for the most kids coding at a single event. "Coding is the language of the 21st century", says Kelly Tagalan from Code Club Australia. "Our mission is to make sure every Australian child has the opportunity to learn how to code, no matter where they live or what school they go to."

Global volunteers

Anyone can start an individual Code Club in their local community, anywhere in the world (see 'How to Start a Code Club' below). There

HOW TO START A CODE CLUB

Code Club is completely free and it's very easy to run your own. All you need is:

- A venue – this could be a local school, library or community centre.
- A volunteer – that's you!
- Children to take part in the club.

Once you are ready to go, all you need to do is register on the Code Club International website and verify that your club is active. You'll get online access to a club account where you can find free resources to use in your club, such as posters, badges, and certificates. And you will be able to use the free curriculum of Scratch, HTML/CSS, and Python projects to teach in your Code Club.

Visit Code Club World: **magpi.cc/2kBa6JL**.

Inspire kids to become the next generation of digital makers

are clubs all around the globe, in countries such as Mauritius, Bermuda, Nigeria, Kazakhstan, the Philippines, and Israel.

Code Clubs come in all shapes and sizes and some of the most powerful Code Clubs make a big difference for a very small number of local kids.

Max Dolgikh runs a Code Club in Ukraine. He's been teaching students to use Scratch. One Ukrainian boy, Vova, wrote a version of the classic game Arkanoid. "I was happy all day long," says Max. "It's worth teaching to just a single child to give them a better life."

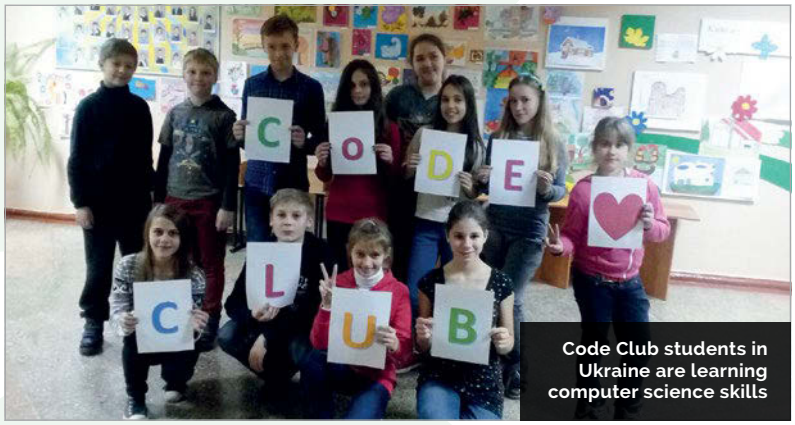
Volunteering at a Code Club is a good way for STEM graduates to pass on coding skills while developing communication skills.

"I have the opportunity to share my knowledge with young people in my community," says Shohel Rana, a Code Club volunteer in Bangladesh. "At the same time, these experiences taught me to communicate concepts to a group, skills that will serve me in the future."

Volunteering is also a good thing to do. "I feel really happy when I am volunteering at Code Club because I feel a sense of real achievement," says Mizanur Rahman, a Code Club volunteer in Bangladesh. "These children do not have big ambitions and they think of doing similar jobs as their poor parents," explains Mizanur. "At Code Club, we provide a creative platform. They can express themselves with confidence and learn skills with better career prospects."

Code Club is looking for volunteers around the world. Anyone can start an individual Code Club in their local community.

Start a Code Club in your local area and you'll get to inspire the next generation of digital makers. Visit Code Club International and don't forget to register your club. Class projects are completely free, and available in 25 languages – from French and Spanish to Korean, Arabic, and Hebrew – for registered Code Clubs. Visit **magpi.cc/2lnuqLL** to see how you can help.



Code Club students in Ukraine are learning computer science skills

CODE CLUB INTERNATIONAL IN NUMBERS

10 OFFICIAL COUNTRIES

9000+ CODE CLUBS WORLDWIDE

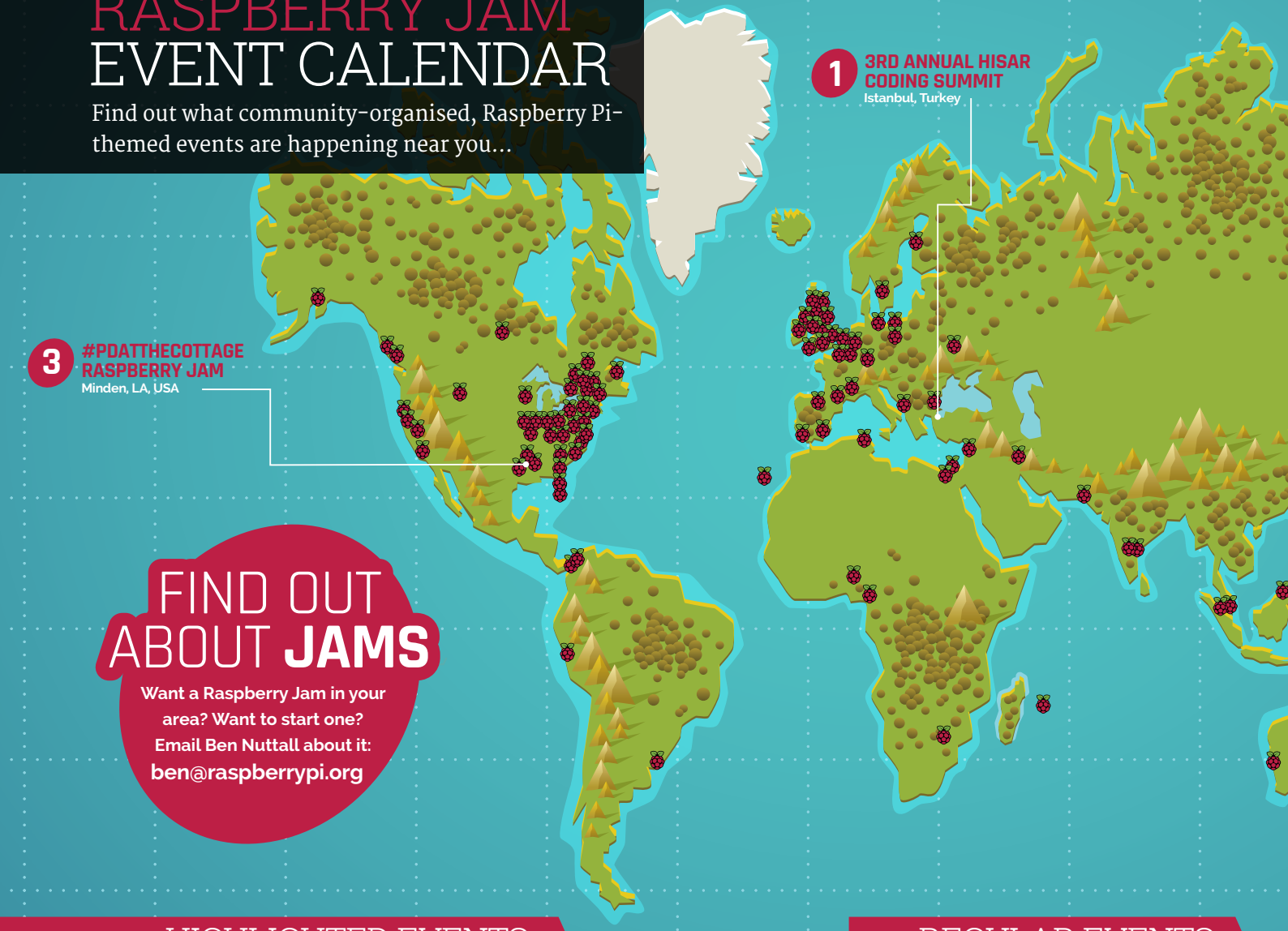
100,000 KIDS HAVE ATTENDED

25+ LANGUAGES FOR PROJECTS

120+ COUNTRIES

RASPBERRY JAM EVENT CALENDAR

Find out what community-organised, Raspberry Pi-themed events are happening near you...



FIND OUT ABOUT **JAMS**

Want a Raspberry Jam in your area? Want to start one? Email Ben Nuttall about it:
ben@raspberrypi.org

HIGHLIGHTED EVENTS

1 3RD ANNUAL HISAR CODING SUMMIT

When: Friday 10 – Saturday 11 March
Where: Hisar Okulları, Istanbul, Turkey
magpi.cc/2ktyM3C
Sharing and spreading essential knowledge of programming and algorithmic thinking.

2 BLAYDON RASPBERRY JAM

When: Saturday 11 March
Where: Blaydon Library, Blaydon, UK
magpi.cc/2ktAFwS
The first Blaydon Jam. If you're interested in Raspberry Pi computers, join in: mingle, learn, share, and enjoy.

3 #PDATTHECOTTAGE RASPBERRY JAM

When: Saturday 25 March
Where: 619 Elm Street, Minden, LA, USA
magpi.cc/2ktjUlM
The inaugural Louisiana Raspberry Jam: discover the world of Raspberry Pi and what it can do.

4 DUNDEE RASPBERRY JAM

When: Saturday 25 March
Where: Dundee MakerSpace, Dundee, UK
magpi.cc/2ktAr9f
Get along to the Dundee MakerSpace and learn about the Raspberry Pi microcomputer!

REGULAR EVENTS

5 PRESTON RASPBERRY JAM

When: Monday 6 March
Where: Media Innovation Studio, Preston, UK
magpi.cc/2lzNExP
A community of people who meet each month to learn, create, and share the potential of the Raspberry Pi computer.

6 MANCHESTER RASPBERRY JAM

When: Saturday 11 March
Where: The Shed, Manchester, UK
magpi.cc/2k476Wn
Bring your Raspberry Pi along and take part in the workshops. There's also a getting started session for newcomers.

JAM HEAT MAP

CAN YOU HELP INCREASE THE AMOUNT OF JAMS IN THE AREAS THAT NEED MORE?

7 STAFFORD RASPBERRY JAM

When: Tuesday 14 March
Where: Stafford Library, Stafford, UK
magpi.cc/2lzwyQu

A big meet-up of Pi enthusiasts who all share ideas, help each other, and most of all have fun!

8 PI WARS

When: Saturday 1 – Sunday 2 April
Where: Computer Laboratory (University of Cambridge), Cambridge, UK
magpi.cc/2k48U1D

Pi Wars is back! The spin-off of the CamJam pits Pi-robots against each other in a series of tricky challenges.

MORE WAYS TO LEARN

CODE CLUBS

The latest figures show there are over 5,000 Code Club after-school or extracurricular clubs around the UK, any of which you can join to learn more about computing. That means there's probably one near you. Code Clubs offer loads of fun ways to learn about computing using Scratch, Python, or even web-style code. Find out where your nearest one is at **codeclub.org.uk**.

YOUR LETTERS

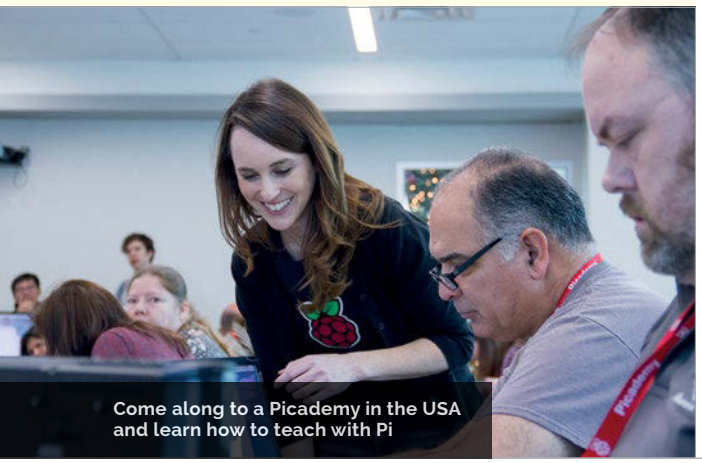
Mature users

I am a very senior (more than 50 years' experience) computer person from the USA. I am also a physician health surveyor for Medicare, so I visit a lot of older folks, most of whom have been bypassed by the computer revolution. My experiments with Raspberry Pi indicate that this, in various form factors, could become a terrific introductory system for seniors who wish to learn about computing (and many of them do!).

This is a HUGE potential application. I note that someone has already cobbled together a simplified user interface for seniors, though I personally would prefer that this be done at the OS level (I know, much easier said than done). On the whole, the Raspberry Pi is just taking off in the USA. It would be terrific if we could get something going with AARP to promote Raspberry Pi for the elderly, especially if we could produce a really EXCELLENT simple but capable user interface.
Stuart Jones

That's great that it's helping seniors learn about computing in America. The Raspberry Pi Foundation's mission does really revolve around children, teens, and other young people, which is probably quite apparent in the way a lot of the resources are written. However, as you've said, the community has come together and created some interfaces more suited to seniors. There's a very long list of stuff that the PIXEL team has to tweak and add, so you may see something like that one day, but who knows when?

Otherwise, it might be an idea to keep an eye on any new Picademy USA courses, so you can learn better how to educate others about the Raspberry Pi.



Come along to a Picademy in the USA and learn how to teach with Pi

Mirror, mirror

Thank you very much for the magic mirror guide in the last issue – it was incredibly informative and I'm getting the components now to try and make my own. I liked the way it was laid out, showing each step in detail – I'm sure not everyone reading a computer magazine will have the necessary carpentry skills; I sure didn't!

Are you going to continue doing these sort of builds at all in the near future?
Tamara O

If you're this far into the issue, you'll probably have already noticed that we do indeed have a similar build this issue with the Pi Zero W retro console. Starting from this issue, we're calling our projects like this MagPi Big Builds – while we have had two in a row, we're likely not going to be doing one every issue, so each Big Build will be a bit of an event. Hopefully, we can keep making projects as popular as the magic mirror!

Above A lot of you liked the big project from last issue and we're definitely planning to do more in the future

Lost in translation

I've noticed you're doing a few translated editions now – the German version and the mini mags. You seem to have a great range of languages; however, I'd love to help you with translating it into another language. Do you have any plans for future translated editions?
Marcus

We do have plans for other translated versions of *The MagPi*, although we can't really talk about which ones just yet until we're absolutely sure they're going to happen. While we do appreciate members of the community offering their services, it's a little tricky in many ways, because we're a magazine and making sure the text is different but the design stays similar requires a lot of extra skills. Do let us know what languages you would like it translated to, though, as we can always try to get that on the plan.

FROM THE FORUM: WHEN ASCII GOES WRONG

The Raspberry Pi Forum is a hotbed of conversations and problem-solving for the community - join in via **raspberrypi.org/forums**

Rather late, I know (or early for 2017): I'm looking at the Christmas tree project. On the image of the back of the tree card, what do the characters mean next to the lights? What does 'FOHN' mean next to the light number? There is also 'FOHN /1', and 'FOHN 05' appears twice.

Thanks,
jonesypeter

This was a weird one, but thanks to the folks on the forums, we did realise what happened...

To begin with, they're supposed to say GPIO and not FOHN. You then wire up the cover to the listed GPIO ports to get them to light in the right order. Simple enough. What seems to have happened is that during the design process, the illustration we used which said GPIO was subject to some bug that changed each letter to the previous one on the alphabet.

Due to the way the articles are checked, it was nearly impossible to notice this error. Now we know it's a thing that might happen, though, we can make sure it doesn't again. We're sorry for any confusion caused!

WRITE TO US

Have you got something you'd like to say?
Get in touch via **magpi@raspberrypi.org** or on The MagPi section of the forum at: **raspberrypi.org/forums**

COMPACT STORAGE & COMPUTING

WD PIDRIVE NODE ZERO FOR RASPBERRY PI

Build your own mini DLNA + Samba file server, wireless mobile storage device, or a Pi MusicBox with local storage and more.

Includes:
- 314GB WD PiDrive, Raspberry Pi Zero, and custom adapter board
- SD card with starter software
- Mini HDMI adapter cable

ONLY $44.95

WDLABS

Learn more at **wdlabs.wd.com/magpi55c**

Western Digital, WD, the WD logo, WDLabs, and the WDLabs logo are registered trademarks or trademarks of Western Digital Corporation or its affiliates in the U.S. and/or other countries. Raspberry Pi is a trademark of the Raspberry Pi Foundation. microSD is a trademark of SD-3C, LLC. The term HDMI is a trademark or registered trademark of HDMI Licensing LLC in the United States and other countries. All other marks are the property of their respective owners. As used for storage capacity, one gigabyte = one billion bytes. Total accessible capacity varies depending on operating environment. Product specifications subject to change without notice. Pictures shown may vary from actual product. © 2017 Western Digital Corporation or its affiliates.

In association with

The PiHut
www.ThePiHut.com

WIN ONE OF TEN KODI EDITION RASPBERRY PI CASES

This issue we've reviewed the brand-new version of Kodi and we've also made a tutorial on how to make your own HTPC with it. This Kodi case is the ideal accessory for your home theatre, blending in perfectly with your other devices.

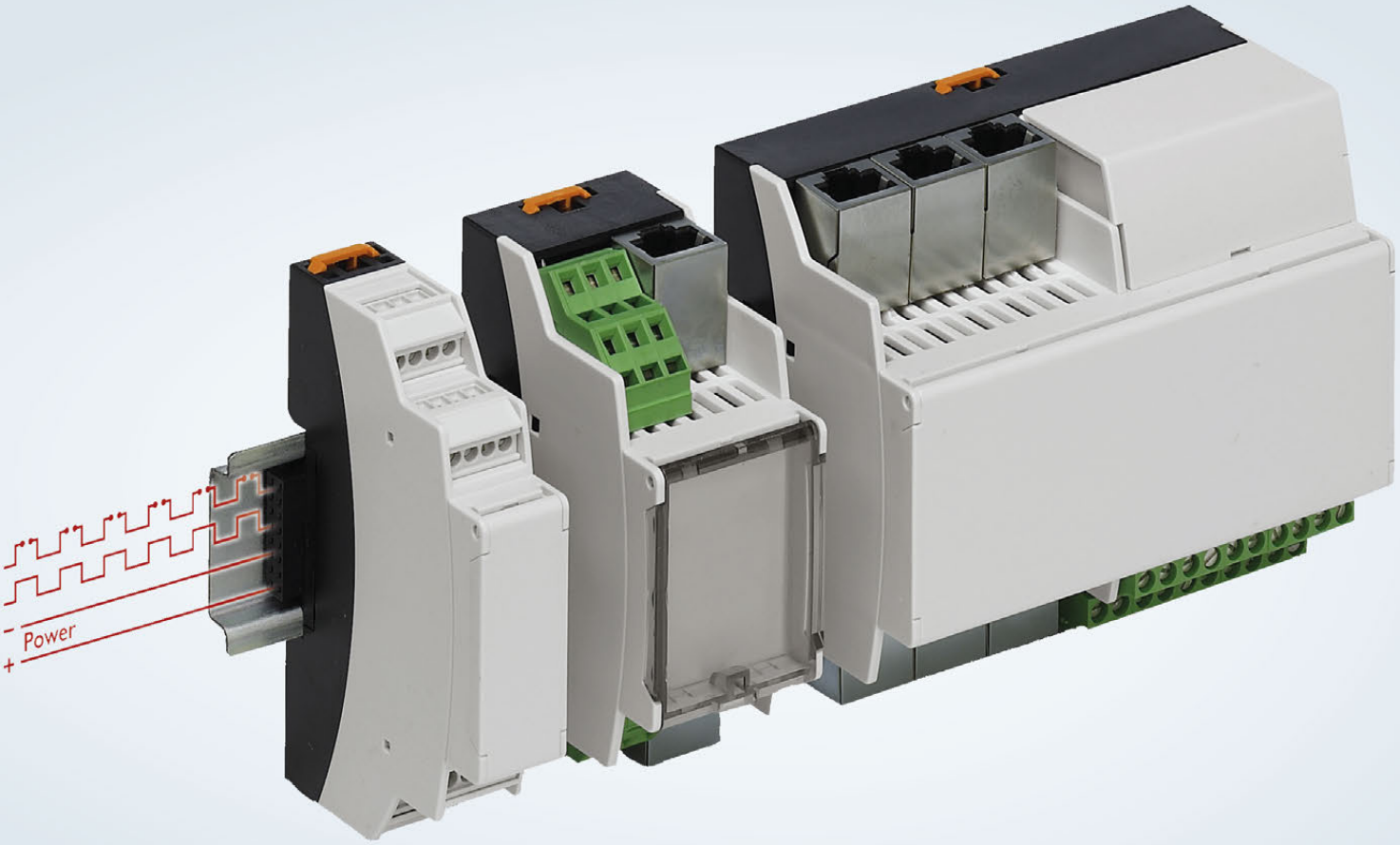
What is the **codename for Kodi 17.0?**

Tell us by 27 March for your chance to win!

Simply email **competition@raspberrypi.org** with your name, address, and answer!

Terms & Conditions

Competition closes on 27 March 2017. Prize is offered worldwide to participants aged 18 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families or friends. Winners will be notified by email after the draw date. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine (unless otherwise stated upon entry). We don't like spam. Participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered.



BC electronics enclosures for building automation applications

The possibilities are endless when you add a Raspberry Pi

The new RPI-BC enclosure for Raspberry Pi computers is part of the well-established BC electronics enclosure range from Phoenix Contact. The BC range is ideally suited to building automation & IoT applications, offering endless possibilities to expand the functionality of Raspberry Pi computers.

For additional information call 0845 881 2222 or visit **phoenixcontact.co.uk/RPIBC**

PHOENIX CONTACT
INSPIRING INNOVATIONS

MATT RICHARDSON

Matt is Raspberry Pi's US-based product evangelist. Before that, he was co-author of *Getting Started with Raspberry Pi* and a contributing editor at *Make:* magazine.

PI FOR THE CONNECTED HOME

Matt Richardson on how the arrival of Zero W is perfectly timed for IoT

Since the original Raspberry Pi Zero came out, I've seen many makers using it for connected home projects. Its size, low price, low power consumption, and software package have made it a great option, even if makers had to use a USB peripheral to add connectivity. Now that wireless LAN and Bluetooth connectivity are built into Raspberry Pi Zero W, it makes this mini computer platform even better suited for home Internet of Things projects.

Let me get this out of the way first: 'Internet of Things', or 'IoT', has all the trappings of an overhyped buzzword. But even if the term Internet of Things doesn't stick around very long, the concept of connected devices is here to stay for good. It's a clear side effect of increasingly affordable wireless connectivity technology.

It's not just development boards that are becoming more connected. The consumer electronics devices that we buy for our homes are more likely to have wireless capabilities. Even a product as simple as a light bulb can be equipped with connectivity, so that you can control its intensity and colour with a mobile app or home automation platform. I recently connected our Google Home to our WeMo Smart Plugs so that I can control the lights in our home using my voice. Last week I was carrying a load of laundry into a dark bedroom. Being able to say "OK Google, turn the bedroom lights on" and having it instantly do just that was a magical moment.

As makers and technology enthusiasts, we have even more power available to us. We benefit from affordable connectivity when it arrives on hardware platforms like Zero W, and can create the connected devices that we hope to see on store shelves one day. We also benefit from being able to interface with consumer-connected devices. For example, a simple hack with a Raspberry Pi lets you use Amazon Dash buttons to control almost anything you want. (Dash buttons are usually used to order a particular product, such as laundry detergent, from Amazon with just a single press.)

Advanced IoT

If you want to go beyond the basics, there are cloud-based platforms that let you manage many devices at once, and create intelligent alerts and actions. Many platforms are already Raspberry Pi-friendly, including the Particle Cloud, Initial State, Cayenne, and Resin.io. Each has its distinct advantages. For example, Initial State makes it really easy for you to create custom web-based dashboards to show you the state of your own sensors and internet-connected devices.

And if you're a beginner, there are platforms that make it easy to get started with connected devices. One in particular is called IFTTT, which stands for 'If This, Then That'. It's an easy-to-use service that lets you connect consumer and maker platforms together without needing to write any code. IFTTT can also go beyond your devices: it can interact with the news, weather, or even local government. In the first partnership of its kind, the City of Louisville, Kentucky recently announced that it's now on IFTTT and sending real-time air quality data, which you can log or use to trigger your own projects. I hope that it's just the beginning for IoT partnerships like these.

With all the recent developments in the Internet of Things realm, Raspberry Pi Zero W comes at the perfect time to offer affordable, portable, and connected computing power. The best part is that exploring IoT doesn't mean that you need to go too far into uncharted territory... it's still the same Raspberry Pi that you already know and love.

The MagPi ESSENTIALS

LEARN | CODE | MAKE

OUT NOW IN PRINT

ONLY **£4/$6**

from

raspberrypi.org/magpi

The MagPi ESSENTIALS From the makers of the official Raspberry Pi magazine

GET THEM DIGITALLY:

Available on the App Store

GET IT ON Google Play

Love your Pi? Love Music?

Designed by audio experts, enjoyed by everyone

10% Discount voucher "MagPi10"

Made in the UK

Jaw dropping audio quality for your Raspberry Pi

Connect Headphones, Speakers, RCA, Toslink, S/PDIF or XLR

We work with all the major music solutions to ensure compatibility

Sonic Pi · moOde audio player · OSMC · piCorePlayer SQUEEZE AUDIO · volumio · roon · Archphile · PiMusicBox · KODI · rune audio · MAX2PLAY

Twitter: @IQ_audio
Email: info@iqaudio.com
Web: www.iqaudio.com

IQaudIO

IQaudio Limited,
Cricklade Wiltshire.
Company No.: 9461908

WANT TO GET NOTICED?

ADVERTISE IN THE MAGPI MAGAZINE

REACH THE RIGHT AUDIENCE FAST

- World's #1 Pi magazine
- Block booking discounts
- Free to download & share
- Live links on iOS & Android

The MagPi is the most exciting mag in tech today, boasting one of the biggest & most engaged audiences in the industry.

FREE DEBIAN + PIXEL LIVE
THE OFFICIAL RASPBERRY PI
NEW RASPBERRY PI ZERO W OUT NOW!
The MagPi
The official Raspberry Pi magazine
RETRO GAMING BLISS
Also inside:
PI ZERO NOW HAS WIRELESS!

FOR MORE INFO EMAIL **MAGPI@RASPBERRYPI.ORG**

Expand your Pi

Stackable expansion boards for the Raspberry Pi

Serial Pi Plus

RS232 serial communication board. Control your Raspberry Pi over RS232 or connect to external serial accessories.

Breakout Pi Plus

The Breakout Pi Plus is a useful and versatile prototyping expansion board for the Raspberry Pi

ADC Differential Pi

8 channel 18 bit analogue to digital converter. I²C address selection allows you to add up to 32 analogue inputs to your Raspberry Pi.

IO Pi Plus

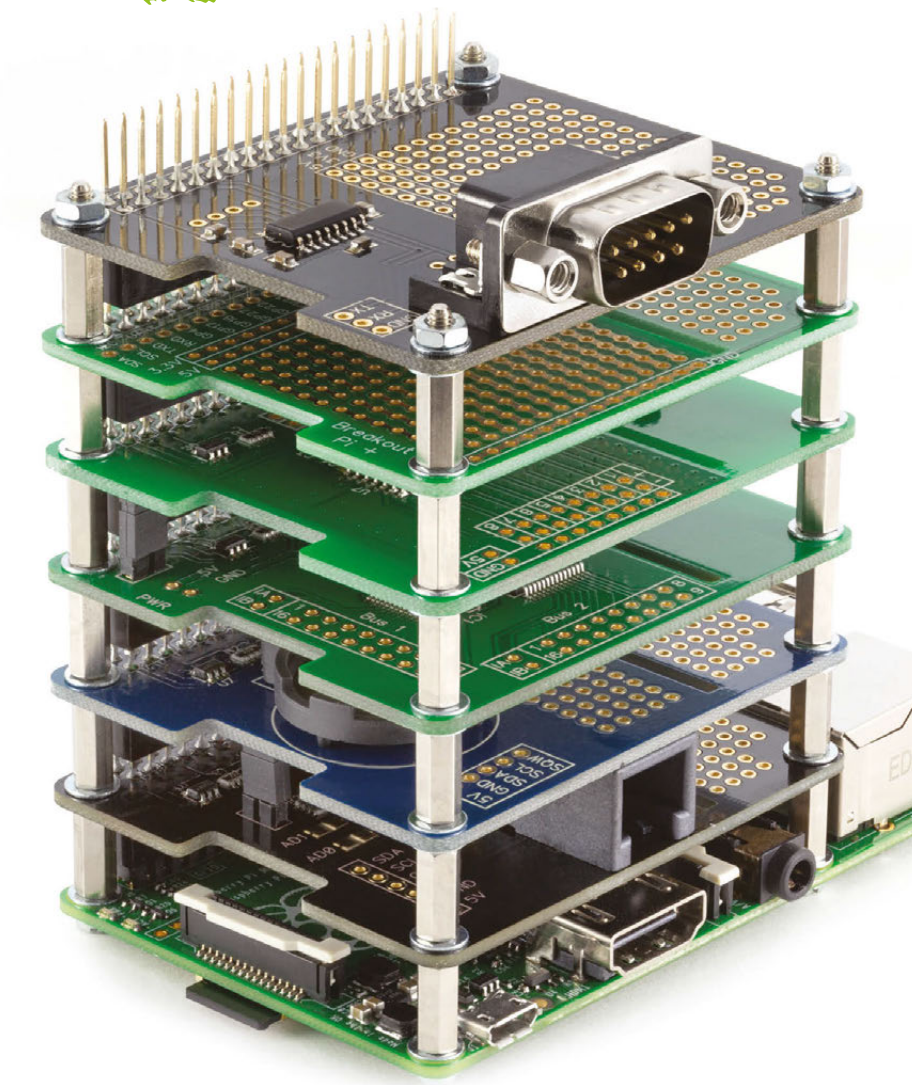
32 digital 5V inputs or outputs. I²C address selection allows you to stack up to 4 IO Pi Plus boards on your Raspberry Pi giving you 128 digital inputs or outputs.

RTC Pi Plus

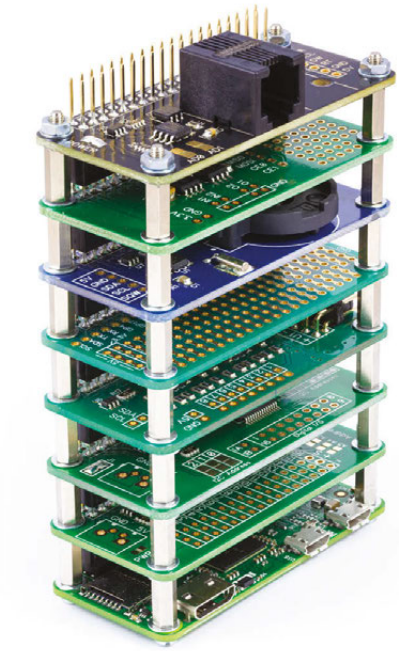
Real-time clock with battery backup and 5V I²C level converter for adding external 5V I²C devices to your Raspberry Pi.

1 Wire Pi Plus

1-Wire® to I²C host interface with ESD protection diode and I²C address selection.



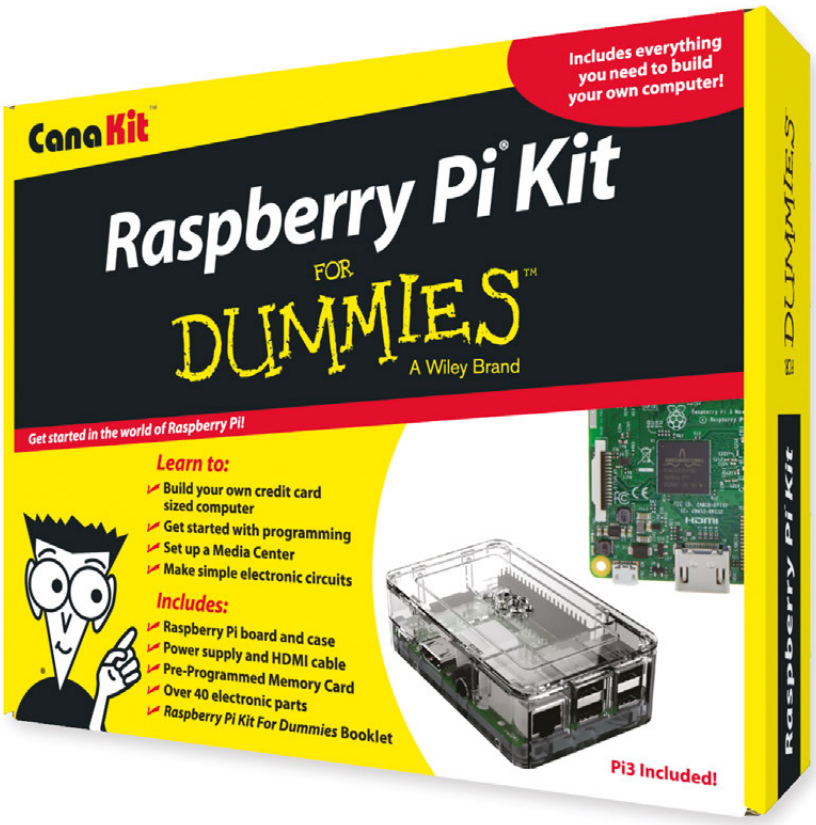
Also available for the **Pi Zero**



ABelectronics UK

www.abelectronics.co.uk

CanaKit™



FOR DUMMIES® A Wiley Brand

Kit Includes:

- *Raspberry Pi For Dummies* Booklet
- Raspberry Pi 3 Board
- Memory Card
- Plastic Case
- 2.5A Power Supply
- HDMI Cable
- Resistors
- LEDs
- Push Button Switches
- Prototyping Breadboard
- Jumper Wires
- Heat Sinks

Available for worldwide shipping at:

WWW.CANAKIT.COM

Available in Europe through RS Components

$89.99 US DOLLARS

£69.99 EXCLUDING VAT

Raspberry Pi is a registered trademark of the Raspberry Pi Foundation. For Dummies and the Dummies Man logo are trademarks or registered trademarks of John Wiley & Sons, Inc. Used under license. RS logo is a registered trademark of RS Components Ltd. CanaKit is a registered trademark of Cana Kit Corporation.

ELECTRONIC KITS • ELECTRONIC PARTS • RASPBERRY PI • ARDUINO